DEFENSE TECHNICAL INFORMATION CENTER

Information for the Defense Community

DTIC® has determined on 11/16/2010 that this Technical Document has the Distribution Statement checked below. The current distribution for this document can be found in the DTIC® Technical Report Database.

☒ **DISTRIBUTION STATEMENT A.** Approved for public release; distribution is unlimited. per NAVAIR 2187

☐ **© COPYRIGHTED**; U.S. Government or Federal Rights License. All other rights and uses except those permitted by copyright law are reserved by the copyright owner.

☐ **DISTRIBUTION STATEMENT B.** Distribution authorized to U.S. Government agencies only (fill in reason) (date of determination). Other requests for this document shall be referred to (insert controlling DoD office)

☐ **DISTRIBUTION STATEMENT C.** Distribution authorized to U.S. Government Agencies and their contractors (fill in reason) (date of determination). Other requests for this document shall be referred to (insert controlling DoD office)

☐ **DISTRIBUTION STATEMENT D.** Distribution authorized to the Department of Defense and U.S. DoD contractors only (fill in reason) (date of determination). Other requests shall be referred to (insert controlling DoD office).

☐ **DISTRIBUTION STATEMENT E.** Distribution authorized to DoD Components only (fill in reason) (date of determination). Other requests shall be referred to (insert controlling DoD office).

☐ **DISTRIBUTION STATEMENT F.** Further dissemination only as directed by (inserting controlling DoD office) (date of determination) or higher DoD authority.

Distribution Statement F is also used when a document does not contain a distribution statement and no distribution statement can be determined.

☐ **DISTRIBUTION STATEMENT X.** Distribution authorized to U.S. Government Agencies and private individuals or enterprises eligible to obtain export-controlled technical data in accordance with DoDD 5230.25; (date of determination). DoD Controlling Office is (insert controlling DoD office).

UCCM Phase II Final report

NAVAIR contract N68335-08-C-0493

Teknowledge Corporation
Principal Investigator: Dr. Allan Terry
aterry@teknowledge.com

September, 2010

Abstract

The BAMS (Broad Area Maritime Surveillance UAS) will be the Navy's next-generation surveillance platform, able to collect far more data than it can send. The data is not of equal value, and the determination of what is valuable dynamically changes during a mission. UCCM (User-Centered Communications Manager) is a software program that sits between the BAMS' different sensors and the radio system to determine what data to send at each moment.

UCCM computes a priority for each potential transmission based on a small number of carefully selected factors. The priority number is a proxy for the current value of that data to the operator, and is dynamically maintained on a priority queue. Each time the radio is ready for a new transmission, it pops the priority queue to get the most valuable data it should send. There is always a default prioritization in place, but the operator can either select items directly (select an image) or indirectly by selecting policies and changing thresholds.

Simulation experiments show that UCCM exhibits many desirable properties. UCCM preferentially sends more recent data first, and sends older items on a bandwidth-available basis. UCCM provides simple means for operators to select images, and avoids downloads of useless images of clouds and fog. UCCM efficiently manages its queue despite potentially overwhelming influxes of readings. Each type of sensor reading has a natural "shelf life." If a reading becomes sufficiently old, UCCM just deletes it. UCCM successfully maximizes the value of data downloaded to the operator, in highly variable situations, using the operator's choices about value.

20101029081

Contents

Background

The UCCM approach to communication management is not tied to the BAMS UAS. It is a general approach to mediating communications between many senders and few receivers, and takes into account problems that arise when the senders and/or receivers are mobile. It was applied in Phase I to the MH-60 helicopter, where many independent shipboard applications send data at the helicopter crew, regardless of what else is going to the crew. In the present Phase II, it is applied to many independent sensors on one aircraft sending data down to a ground station.

The BAMS Communication Problem

The BAMS will be an airliner-sized UAV with a payload of at least 3,000 pounds. It will carry significant onboard computing power and storage. There should be storage for at least several hours of data. It will cruise at 310 knots. BAMS has a ceiling of around 60,000 feet but can come down to as low as 1,000 feet for close observations. Missions can last up to six hours. The prime contractor for BAMS is Northrup Grumman.

BAMS will have a crew of four at a ground station: a pilot, a Communications Officer, and two sensor operators. The Communications Officer is in charge of ensuring that the data flows smoothly and that the needs of the various customers for that data are met. This person is UCCM's primary user. Data requests come from the sensor operators and from external data customers. All three operators have significant domain expertise and are able to make many targeting and interpretation decisions on the spot.

BAMS will have three radio channels available for data transmission. Other channels will handle guidance and telemetry. Interaction between UCCM and the operators is explicitly carried on a flight-control channel and does not subtract from bandwidth on the data channels. The Communications Officer assigns sensors to channels and monitors the dynamically changing bandwidth. This is one of the central problems for BAMS: the bandwidth on any channel varies from moment to moment, and can go to zero. When a channel's bandwidth declines too much, the Communications Officer can reassign some or all of its sensors to other channels, throttle back collection, or choose to let UCCM accumulate transmissions, knowing that the top of the queue will be the most valuable.

A Ku-band satellite channel can carry up to 10 Mbps when everything is perfect. More often it provides 1–3 Mbps. This band suffers from rain fade. It is also possible for the relay satellite to be in a bad location, or for satellite time to not be available. (The operator can use either military satellites or commercial ones.) A C-band channel is available. This is a lower-power, line-of-sight radio with lower bandwidth than the Ku band. A VHF channel is also available, running at modem-range speeds.

BAMS Sensor Package

The primary sensor is the radar, with all its different modes. Perhaps secondary are the EO camera (Electro-Optical, otherwise known as visual) and the infrared (IR) camera. There are also a number of "signals" sensors.

The radar's five modes are mutually exclusive. It can be run in search mode or imagery mode. Search mode is what one normally thinks of as radar—it sweeps while transmitting pulses, and records what is reflected back. A sweep takes 5–20 seconds. A slower sweep for a

given sector trades time for more accurate data. The operator can control the sweep from a full 360^0 to any angular sector. The operator can also control the intensity threshold for returns. A lower threshold means more data, but also noisier data.

The default mode for the radar is search mode, analyzed into tracks as output. Search mode is tactical—it tells the operator where things are, but not what they are. The BAMS is able to fuse multiple returns into a connected track. Sometimes it can merge the radar track with AIS data to get an identified track. There are probably not as many tracks as raw returns, but a significant percentage will be fused into tracks. Radar tracks are updated in each sweep and transmitted. A radar track contains a track ID and some representation of the path with current speed, and current heading of the object. We assume a radar track takes 1–5 KB.

The track fusion algorithm is not perfect, if nothing else because the data is uncertain and noisy. The object being observed is often in motion, and the BAMS is moving at over 300 mph in some other direction. The operator may set the radar into "raw returns" mode. Instead of tracks, the radar downloads the returns themselves for a traditional "paint the screen with each sweep" display. Returns data contains a lot of junk. Entities will appear to bounce around from one sweep to the next. A raw return has some identification key, a direction in spherical coordinates, an intensity value, a phase value, and possibly a polarization. We assume a return is about 1 KB of data.

The radar has three imagery modes: SAR, ISAR, and HRR. All return a grayscale image that we assume has 8-bit color depth. The operator can aim the radar to get an image of a specific area, and can select a resolution less than the maximum. The maximum resolution is an image of 3,000 pixels on a side.

SAR is synthetic aperture radar. SAR takes multiple radar images as the BAMS and target move relative to each other. It then creates a higher-resolution image than is possible for one image in isolation. Like long-baseline interferometry in astronomy, the multiple images are merged as if they were a single image taken from one very large aperture radar. The synthesis process takes about 10 seconds of data and processing. The image is basically a height map, but looks much like a grayscale visual image of the same scene.

ISAR is inverse SAR. This algorithm uses the Doppler histories of the scattering centers in the target area, so the radar is concerned not only with the intensity of the return but also the frequency. The resulting image is used to see what things are moving, and some indication of how fast. An ISAR image requires about 30 seconds of data and processing.

HRR is high-resolution radar, and can be thought of as a strip of an ISAR image. The image is full width, but only a small number of pixels tall. The image is used in isolation, to look at some specific feature. There is no expectation that it will be followed by adjoining strips until a full *NxM* image is built up. Its advantage is that it is fast to collect. This is an unusual mode, so the expectation is that if the operator asks for it, they are looking for something specific and expect high priority.

The EO sensor provides a traditional RGB color image with 24-bit color depth; the IR sensor returns 12-bit grayscale. In both cases, the maximum image is 3,000 pixels on a side. The operator can pan, zoom, and choose to collect less than the full image. These cameras can be used in snapshot mode and can collect (very large) images as fast as the operator can snap them. A video mode is also available, where snapshots are automatically taken at regular intervals. We assume the operator would step down the resolution to something like SVGA

size, and only a few frames per second. Since this mode is very bandwidth intensive, it will be used for only short periods of time, and the operator will expect high priority for the data.

There are two other sensors. The AIS (Automatic Identification System) sensor identifies ships and their movements. The ESM (Electronic Support Measures) sensor identifies radio and microwave emitters in the area. Both are omnidirectional receivers. We loosely classify AIS, ESM, tracks, and raw radar returns as "signals sensors," as opposed to the various image sensors. The signals sensors are all characterized by high rates of very small readings. They are all passive collectors that are on constantly. The image sensors are mainly aimed at specific areas, produce readings that are many orders of magnitude larger than the signals sensors, but are collected less frequently.

AIS is based on a transponder required to be carried by all ships over 300 tons. AIS is based on a short-range radio system whose goal is to inform nearby ships of identity and movement. For example, this is very useful for collision avoidance within a harbor. Normally, AIS is heard horizontally only to 50 miles or less, but vertically has been picked up by satellites. There are a number of AIS messages, but all are less than 1,000 bits in length. There is a periodic broadcast of location, heading, and speed at a frequency that depends on the current speed of the vessel. Each vessel also broadcasts a static message every six minutes that identifies the ship, its characteristics, its destination, and current ETA. The dynamic messages are sent according to the schedule in Table 1. Needless to say, military and hostile ships generally turn off AIS when on a mission.

Frequency	Under which conditions
Every 3 minutes	When at anchor
Every 10 seconds	Moving at 1–14 knots
Every 3.33 seconds	Moving at 1–14 knots but changing course
Every 6 seconds	Moving at 14–23 knots
Every 2 seconds	Moving at 14–23 knots but changing course
Every 2 seconds	Moving faster than 23 knots

Table 1: Schedule of AIS Messages

ESM listens in a wide band of radio frequencies for emitters. It picks up radar, radio communications, cell phones, and most other emitters in the selected frequency range. ESM readings are small—they are basically a direction in spherical coordinates, a frequency or band, and a few characteristics of the reading. However, ESM flux can be very high over an active area. Imagine flying near a city full of cell-phone users. ESM is useful to provide some notion of what emission sources are in the area. Their type and distribution is an indicator of the kinds of activity going on in that area. Considering that military systems stay off the air as much as possible, use frequency hopping, and many other forms of deception, the data is not as definitive as that from other sensors.

The vigilant may object that listening to ESM at the same time as operating the radar is a bad combination. The radar sends out radio frequency pulses, and ESM would just pick up what is reflected. The sensors use "blanking," which effectively interleaves the two sensors at the microsecond level. ESM listens in the short intervals between pulses where returns are not expected.

Finally, some of the literature mentions a possible data-relay mission for the BAMS. In a line-of-sight world, a mobile, high-altitude relay point can be very attractive. We interpret this mission to be outside the scope of UCCM. Data relay is likely to work by carving out a portion of a channel and dedicating it to voice or other communications. Since it is somewhat like a virtual private network over this channel, it won't be prioritized. The only impact on UCCM is that the channel will appear to lose some bandwidth for as long as the relay is active.

Sensor Usage

The radar is always on, in one of its modes. Radar can be used from any altitude but is the main sensor at the higher altitudes. Radar has the advantage that it can see through clouds and fog.

The EO and IR sensors are mainly useful below the cloud deck, say, 10,000'. It is often said there are clouds 50% of the time in a marine environment. Low altitude cumulus clouds form at 4,000 to 10,000', while altostratus clouds can form up to 20,000'. The IR sensor is particularly useful at night, but can be used in some situations during the day.

Imagery can be requested by the pilot to help steer the aircraft and to help make immediate decisions such as where to point a sensor next. It can be used by the sensor operators to do preliminary analysis and then decide what else to collect. Or it can be treated as intelligence, to be used by some external information customer at a later time.

BAMS computes radar tracks on board. It does not wait for significant change; it just sends the track each time a point is added. It is also possible to have AIS-based tracks, but they are not computed on board. The operator will want to see the most recent AIS messages first, but the earlier ones are useful for backfilling where the vessel has come from. They should not be considered as being superseded by more recent information. Older tracks can be superseded because the newer track contains the previous information.

When the operator asks for something unusual, that can be considered to be a signal of their intent—they want the data now. Asking for a video is an example. The operator knows this will consume most of the bandwidth, and a video requires successive frames in something like real time to be useful. The ground station could buffer images and replay the video later, but the operator requests video for its immediacy. Asking for HRR is another example. It is an odd format. The operator will request it as a faster alternative to ISAR, and since it is so narrow, they must be looking for something specific. They will want the data immediately. Switching to raw returns from tracks isn't quite as unusual, but still rates some priority. It says the operator does not quite trust the generated tracks, and is willing to swim in a sea of noisy data in hopes the different perspective will show something. They will do this to make an immediate decision; this data is rarely useful to save for later analysis.

The operators can and do take responsibility for managing the data flow. The job of UCCM is to make this as painless as possible. In particular, most sensors have parameters and thresholds the operator can use to trade off the volume of data against sensitivity. The operator can throttle the data flow to match the available bandwidth. UCCM tries to implement or even anticipate the operator's intent, and to ensure that only the most useful data gets sent.

The goal of UCCM is not to send all the data that is collected. The goal is to send the most valuable information possible, given the circumstances. Quantity of data is not the issue. Sending images of cloud banks is not particularly useful, and the sensor package can collect far

more data than is possible to send. UCCM needs to infer what will be most useful to the operator, and to make sure that gets sent out before anything else.

Use Scenarios

The primary mission for BAMS is classic general surveillance. It is sent to an area to see what is there and what is going on. It may be sent to look for something specific. A second mission is to collect intelligence for later analysis.

We are using surveillance of Somali pirates as a use case. The mission shown in Figure 1 is an example. The BAMS patrols west along the Gulf of Aden in leg 1, looking for merchants and naval ships. It looks for pirates near merchant ships but out of view of them, and opportunistically elsewhere. For leg 2, it flies back east along the coast of Puntland (of northern Somalia). The crew watches for pirates, but the main objective for this leg is to scan the coast for bases. At the end of the mission, in the evening and night, an intelligence agency requested leg 3—several orbits of the island of Socotra, with emphasis on the Haghier Mountains around the villages of Jo'oh and Ar Rak.

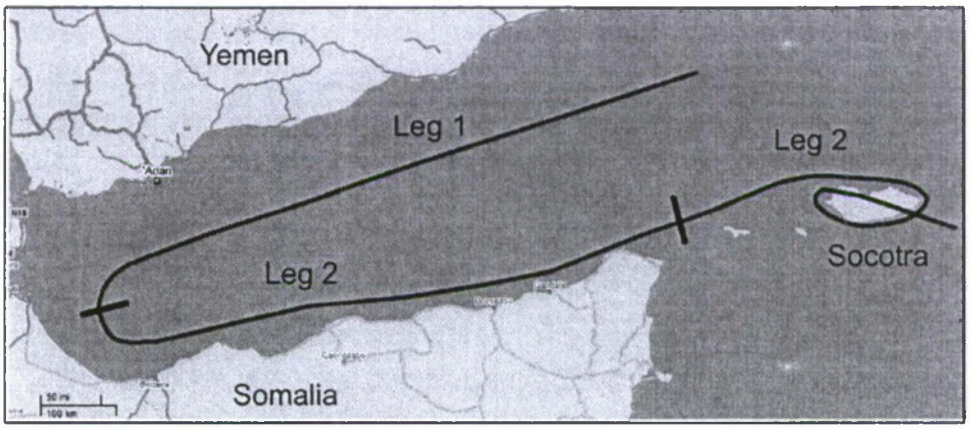


Figure 1: Somali Pirates Use Case

Leg 1 covers an area that is 500–600 miles long. It will take 2–3 hours, depending on events. The BAMS will cruise at 350 mph at 25,000', using radar ranging to find items worth investigating. Satellite bandwidth at the start of the mission is 1,200 Kbps. There is a cloud deck at 12,000-15,000'. Early on, BAMS finds a tanker and descends to 10,000' to search for small, unidentified vessels in that area. It finds none, but as it climbs back to altitude, it finds a second merchant on the horizon, but also loses some bandwidth. While looking around this second merchant, the sensor operator sees a new blip that could be a small ship. This other ship is about 40 miles away from the merchant, but in its path. The radar track is iffy, so the operator switches to raw data mode for a quick double check. This looks sufficiently interesting, and a flyover produces a good visual shot of a mother ship towing two launches. The crew notifies the Combined Task Force, which alerts the merchant.

Leg 2 cruises just off the Somali coast at 8,000' and should take a few hours. The goal is general intelligence, and the customer wants good visual imagery. The crew alternates radar with ESM to see what else they can learn. Along the way, the C-band channel drops out and the satellite bandwidth fades to 400 Kbps. The crew finds what may be a new pirate base with

associated construction a few miles inland. The crew snaps a number of good shots and marks them as held for later sending—they are not needed in real time.

Towards evening, the BAMS starts leg 3. Not much is expected of immediate Naval interest on the island of Socotra, but an intelligence customer is very interested in a couple of specific areas in the mountains. Socotra is only about 50 miles long, so the crew does several orbits, starting at 10,000' with EO and IR. After several ascending orbits, the BAMS is up to 40,000' and starts surveying with SAR.

Expected Data Rates

The Somali scenario actually represents light data collection. The number of TRs sent is dominated by AIS, ESM, and Tracks. The number of bits sent is generally dominated by images. The Somali mission has a low rate for AIS and ESM, especially when flying over open ocean.

Table 2 describes our current knowledge of TR arrival rates. The worst-case scenario describes a situation such as flying over a large, busy harbor and tuning ESM into a cell-phone frequency. The moderate scenario covers a situation where BAMS is busy due to interesting situations on the ground, but of a more routine nature. A light scenario (not shown) would cover situations such as observation of a shipping lane at sea or a third-world coastline.

Content type	Moderate arrival rate	Worst-case arrival rate	TR size (KB)
AIS	10 / sec.	20 / sec.	0.1
ESM	400 / sec.	1,000 / sec.	1.0
Tracks	7 / sec.	25 / sec.	2–5
Raw radar returns	50 / sec.	200 / sec.	1.0
HRR image	1 / min 0.017 / sec.	10 / min 0.167 / sec.	8-bit grayscale 50–200
SAR image	1 / min	3 / min 0.05 / sec.	8-bit grayscale 1,000–8,800
ISAR image	1 / min	2 / min 0.033 / sec.	8-bit grayscale 1,000–8,800
IR image	1 / min	6 / min 0.1 / sec.	12-bit grayscale 1,000–13,000
Visual image	1 / min	6 / min 0.1 / sec.	24-bit RGB 3,000–26,000
Combined rate	468 / sec. or 28,000 / min.	1245 / sec. Or 75,000 / min.	

Table 2: Data Rate Assumptions

The Value of Data

The primary user for UCCM is the Communications Officer, who tries to ensure that the different data customers get all of the data they ask for, in the most timely way possible. The ultimate goal of UCCM is to ensure that the operator always gets the most valuable data at any given point. Value is a subjective measure that only the specific operator can define. The data that the operator needs and values most changes instant by instant as a mission unfolds.

What makes a given sensor reading valuable? A reading can be used in real time for the operator to make a decision. An image or a track might be used to decide where to steer the

aircraft or identify a target for follow-up. In these cases, it is obviously important to get the data as quickly as possible. Data can be valuable because it confirms a hypothesis or reveals something new and interesting. The data in this case may or may not be needed in real time; it could be used in analysis later. Certain sensor modes are used basically for "second opinions" only: HRR imagery and raw returns. Again, getting the right data at the right time is key. Other readings are valuable for what they are. AIS can provide exact identity. ISAR provides a movement-based viewpoint. UCCM's priority calculation tries to capture all the different perspectives. UCCM can't directly ask the operator to put a value on each reading, so it tries to model the operator's needs as closely as possible.

Priority is used to decide which TR in the UCCM queue will be sent next, and sometimes which to delete. TRs are sent in strict arrival order in the FIFO queue and priority plays no role in the TR's life cycle. However, FIFO TRs are prioritized exactly the same way as UCCM TRs. The priority number is the proxy for the value of the TR, and is used to compare the behavior of the two queues on a like-kind basis.

Metrics of Quality

UCCM tracks three quality metrics (as opposed to performance metrics). For the purposes of this SBIR, UCCM results are compared to those from a simple FIFO regime.

Cumulative average priority of the transmitted TRs

This is a simple arithmetic average of all TRs sent since the start of the mission, one average per radio. There is a screen that displays these averages in real time. This is the basic "how are we doing?" score. Clearly, we expect the UCCM curve to show a higher average priority than the FIFO curve. One drawback is that as thousands of data points are collected, it becomes harder and harder for new data to move the curve significantly. It becomes less a measure of performance at a point of time and more a measure of total mission performance.

Scatter plot of the priority of TRs when sent versus their age

The age of a TR is the number of minutes it has spent on the queue. The ideal state is that high-priority items get sent more quickly than low-priority items. This chart can reveal other details of a session; its interpretation depends on the bandwidth profile and other characteristics.

The "Negabits ratio"
This metric is modeled after the Negawatts of energy conservation. The best kilowatt is the one you never even generated due to efficiency, reuse, and better design. The best use of the aircraft's limited bandwidth is to send only the best data and not use it for low-value data. The negabits ratio is the total number of MB that UCCM explicitly decided not to send, divided by the number of MB it did send. Items counted as not sent include: images the operator deleted or held after seeing their thumbnail, all held TRs, TRs deleted because they became too old, and images still held in the image buffer because the operator has not acted on them. Bigger is generally better, but interpretation is still contextual. A low ratio can arise when the bandwidth is high, or when there is a high density of useful data. It rises higher when there are a lot of images in the mix, when a high flux of TRs makes aging more important, or when the operator rejects most images for being of low utility. A high negabits ratio means UCCM needed to apply its heuristics more in order to deliver value.

UCCM Architecture

UCCM is functionally the gateway to the aircraft's radio system. Transmission requests (TRs) arrive from any sensor that wants to send a reading. UCCM maintains a priority queue such that when the radio is done with its previous transmission, it will start sending the highest priority item on the queue.

UCCM is architected as a simple Web application. The aircraft runs the server and its database. The client in the architecture is the operator's station on the ground. Communications in this version are done via the HTTP protocol. UCCM is implemented as an application on top of the Teknowledge ActionWeb™ platform. ActionWeb provides an engine for rule-based programming (including inference), ability to define connectors that bring sensor data into the system, ability to define actions that the rules call, and an extensive development environment. ActionWeb integrates several best-of-breed open-source components into a useful whole that supports observe-deliberate-act agents. The implementation is a basic Java 5 EE using POJOs (simple objects) rather than Enterprise Java Beans. UCCM uses a RAM-based SQL database.

UCCM is structured for development as shown in Figure 2. The application itself is embedded within a simulation of the aircraft. This simulation is discussed below, in the Simulation Experiments section.

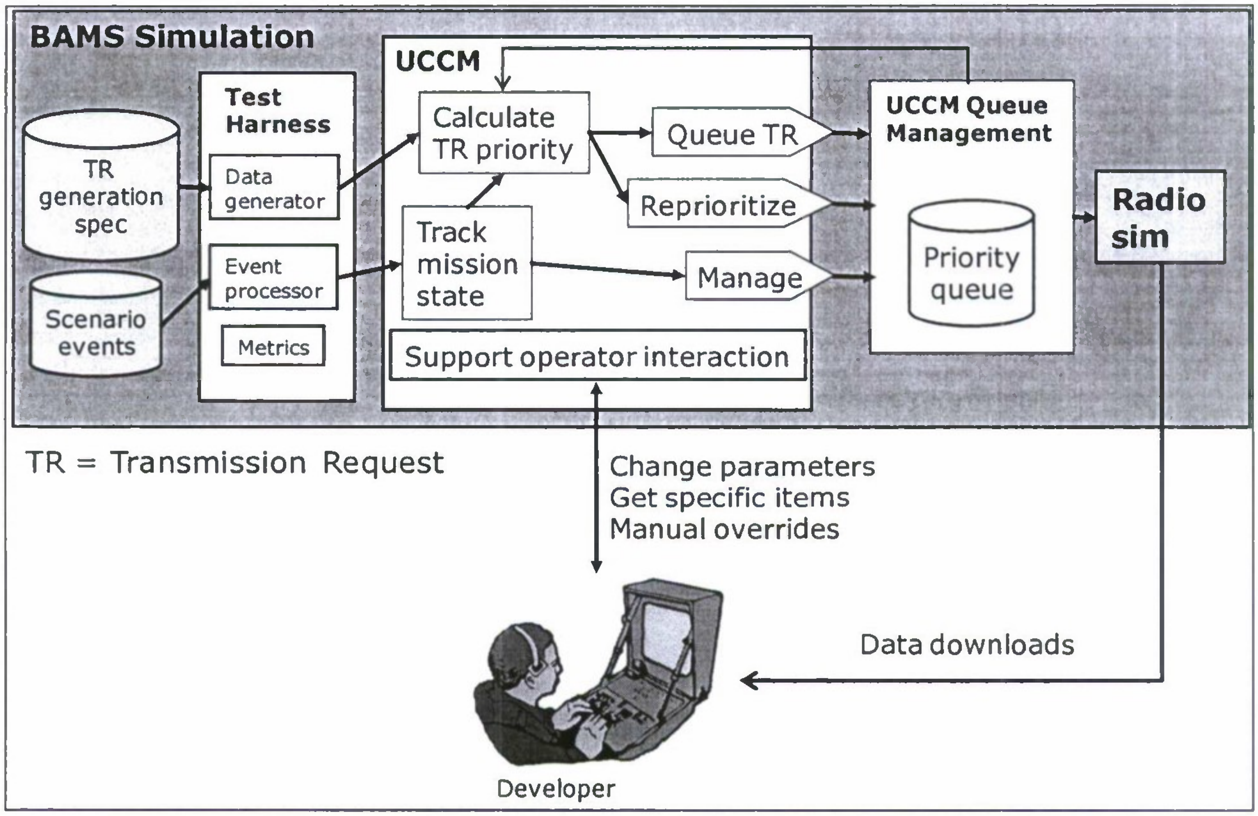


Figure 2: UCCM Development Architecture

UCCM proper contains two components: a rule engine and a queue manager. Prioritization and event-driven computation is handled by the forward-chaining rule engine. The rules deliver prioritized TRs (PTRs) to the queue manager. This Java component manages the queues, enforces priority thresholds, implements the "hold" mechanism, selects PTRs to send back for reprioritization, and calculates metrics.

Figure 3 shows UCCM in a deployment context. UCCM is integrated with the aircraft instead of a simulation. The main change is that TRs come from the BAMS sensors through ActionWeb data-driven connectors. Each connector accepts a reading and processes it into an ActionWeb event. In this case, the events are mostly TRs, with a much smaller number of operator actions. Other connectors can bring in data for self-management, such as current operating parameters, the aircraft's heading, the status of the three channels, and so forth. The UCCM connectors for sensor data will also implement the bundling of many individual sensor readings into one TR.

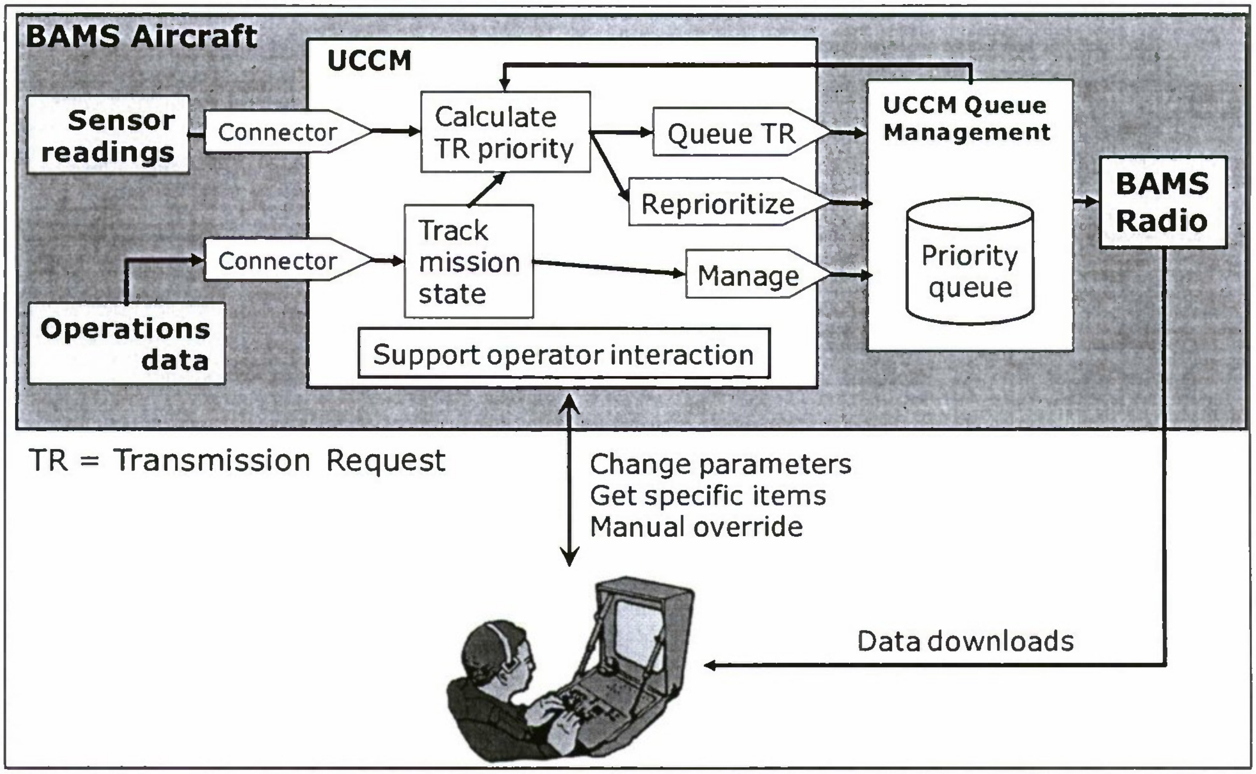


Figure 3: UCCM Deployment Architecture

Interaction with users will take place in the rule-based component. GUI interactions are typically reactive, and a rules implementation is natural for that style of programming. User input can also be treated as similar to sensor input. It changes the state and how prioritization occurs from that point onward.

UCCM itself is not a hard real-time system since it is written in standard Java, but fits as a component within such a system. Sensor-reading inputs into UCCM go into an event queue that buffers UCCM from the variable rate of inputs from the aircraft. The prioritized transmission queue also buffers between UCCM and the radio subsystem. The radio just needs to pop the queue when it is ready to send the next transmission. UCCM ensures that the queue is filled and correctly prioritized. The function UCCM provides is not on avionics or other aircraft critical paths. Its users are humans on the ground. As long as they perceive that they are getting the data they need in a timely way, hard millisecond time budgets are not required.

The Prioritization Algorithm

The transmission request priority is the basic data that runs UCCM. The rules compute them and the queue component acts on them. The priority calculation is a weighted linear combination of a small number of factors. It is a trade-off between high domain precision and practical utility. Considering a large number of factors may capture all nuances of a situation, but at the cost of collecting them and maintaining the knowledge that employs them. Once more than a small handful of factors are considered, no one factor has the ability to make much difference in the final prioritization. UCCM uses two static factors (size and importance) and

two dynamic factors (TR age and operator's intent). Our tests to date show that these factors provide a workable and useful compromise.

Each factor should compute a score from -100 to 100. Each factor should compute its score with no regard to other factors; it should be an independent score. There will be a heuristic cutoff, a parameter, currently between 20 and 50. TRs below that level of priority will not be deleted, but neither will they be sent. They will remain on the transmission queue until explicitly deleted, or until conditions change that increase their priority over the lower limit. There is another parameter that is the lowest acceptable priority. Any TR with a priority below this limit will be deleted.

The fundamental issue is that there isn't enough metadata for UCCM to make very many decisions about utility. UCCM does not attempt to look inside the content of a TR. In other domains, such as helicopter communications, there are factors such as whether a TR is related to a current or future leg of the mission, the time before a decision is forced/required (by an approaching aircraft that might not be friendly, for instance), the "speech act" of a communication, or the importance/status of the sender of the communication. The connectors that bring sensor readings into UCCM add the metadata listed in Table 3.

TR Metadata item	Description
contentType	A tag, such as AIS, Visual image, or Track.
Size	The size, in KB, after compression is applied.
arrivalTime	Time stamp from the mission clock.
age	Age, in minutes. Computed when needed.
radarType	Boolean. A convenience classification.
intent	if one has been specified. Otherwise and more often "Not specified." Values include selected, held, deleted, thumbnail, and image buffer.
numberInBundle	Some TRs represent a bundle of TRs of the same type.

Table 3: TR Metadata

Transmission Request Preprocessing

The ActionWeb connector for each sensor will be able to compress the TR payload before inserting the TR into the system. We assume the operator will have several alternatives for compression, and that they will be settable as policy or configuration decisions. For the sake of this project, we assumed AIS will not be compressed since it is already binary encoded and very small. All other TRs are compressed to 25% of their original size. The compression factors used in a deployment are likely to be different, and could change during a mission if different compression algorithms are selected.

The preprocessing rules set various convenience fields that classify the TR. For example, whether it is a TR type from the radar or not, or whether it is an image that requires a thumbnail.

For the sake of Phase II experiments, TRs may be forked into multiple copies. A TR is put on either the UCCM queue, the FIFO queue, or "Both." The UCCM queue treats many of its TRs differently than those on the FIFO queue. When the arriving TR will have a different life history,

it is forked into separate copies for the different queues. They start out with the same description and metadata but thereafter may evolve in a completely independent manner. A few examples of forking rules:

- When an arriving TR has intent = held, it is forked into a UCCM copy and a FIFO copy. The UCCM copy will be subject to the held-TR rules, while the other copy is subject only to first-in, first-out selection.
- An image that arrives with intent = "not specified" is forked into three. A copy of the image is put on the FIFO queue. The original image is not queued to be sent, but put into an image buffer. Finally, a small thumbnail is created, and that is put on the UCCM queue.
- TRs that get special treatment (such as HRR images or radar returns) are forked into UCCM and FIFO versions.
- TRs that have intent = "not specified" and are not handled elsewhere are not forked. They are queued to Both, basically meaning the same TR goes on both queues but still in an independent way.

This notion of forking is clearly an experimental convenience and would not be used in a deployed system that does not run the FIFO queue.

The current BAMS algorithm uses several mechanisms to increase the flow of value in addition to simple priority:

- The image-thumbnail framework directly collects intent from the operator.
- Aging implements the intuition that any specific reading from the signals sensors gets stale quickly, and even images get stale if not sent in a timely way.
- A way is provided for the operator to boost all readings for a sensor.
- Several means are provided to delete TRs before they are sent, or hold them to be downloaded upon landing.

Only then are straight priorities used.

Priority Due to Size

UCCM can note current bandwidth along with TR size, but only uses size in the calculation. With both, one could compute the number of seconds needed to transmit the TR and use that in calculations about "filling the pipe." We chose to only use size because size is an intrinsic property, while bandwidth changes dynamically. Using bandwidth would require reprioritization whenever the bandwidth changed in a material way. When bandwidth changes, all TRs remain in the same relationship to each other, but the queue gets cleared faster or slower.

The smallest TRs are AIS readings of around 0.1 KB; others are around 1 KB. It can easily happen that it costs more to prioritize the TR than to just send it. For this reason and for performance reasons, we assume the relevant ActionWeb connector will batch a number of readings from certain sensors into a larger bundle. For example, accumulating the next *N* readings, or all the readings within one second. The ground station will unpack the bundles into their constituent individuals before use. We will bundle AIS, ESM, tracks, and radar returns. ESM and returns can easily be bundled because the individual TRs lack specific identity and are not reasoned about as individuals. AIS messages could be reasoned about as individuals, but we chose not to. The operator could maintain a "white list" of known ships

they don't need updates for. That filtering is a simple function the ActionWeb connector could provide. There is no need to do it in rules. Tracks have an identity to the extent of a track number. "Whatever this track represents, here is its update." Since one track update contains previous positions along the track, a newly arrived track can supersede a previous copy. However, for performance reasons, UCCM will bundle tracks and trust to the aging mechanism to handle the case of one track superceeding older versions. UCCM will not bundle images.

The result of bundling is that the smallest TRs are held to a size of 1–10 KB instead of 0.1 KB. When bundles are several seconds long and there is a high arrival rate, bundles will more often be in the 50–500 KB range. Since the TR arrival rate is dominated by the signals sensors, bundling can cut this flux significantly. As an extreme example, the ESM sensor can burst up to 1,000 readings per second, but UCCM need only process one 1-second bundle.

In deriving a score for TR size, we consider the range of sizes. The small TRs of 1 KB should get a score close to 100. A TR of average size should score in the 60–70s. It should go to zero about at the cutoff for the largest TR UCCM will send (currently 2,000 KB). The largest TRs will get zero or even negative scores. The curve for priority due to size should show the most discriminative power between small thumbnail size and average image size. This argues for a sigmoid curve. We selected:

$$\text{Priority} = 100 - \frac{scale}{1 + e^{(-steepness \cdot \log(size) - \log(inflection\,point))}}$$

size is the size of the TR, in KB.

The *scale* factor allows priorities to go below zero. Priorities range from 100 to (100–*scale)*.

inflectionPt is the size value for the curve's inflection point, and

steepness is a factor that determines the width of the peak.

If a channel contains the full range of sensors, TR size ranges from 1 KB to 20,000 KB or more. The small TRs (AIS, ESM, radar returns, and tracks) will be virtually indistinguishable at this scale. The function needs to clearly distinguish the small TR group, the thumbnails, small images, and large images. UCCM uses *scale* = 200, *steepness* = 1.5, and *inflectionPt* = 2,000 KB. This curve is shown in Figure 4.

Several factors will temper this calculation. First, the smaller content types will come in bundles of readings that are 10 to 100 times bigger than individual TRs. This boosts their size into the region this curve can distinguish. Second, we assume compression will be used before UCCM gets the TR. A thumbnail could use lossy JPG compression to significantly reduce the size, while its full image might use a lossless compression once it is known that it is worth downloading. If JPEG-style compression is allowed, with only moderate compression, the 26 MB image might compress down to 2.5 MB or so.

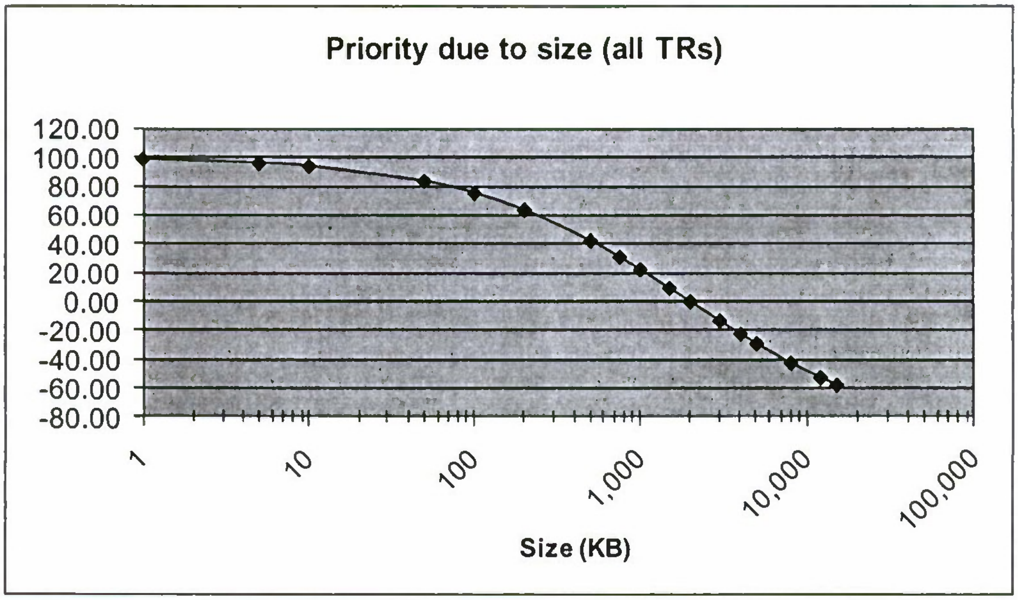


Figure 4: Priority Due to TR Size

When a channel has no image sensors assigned to it, it may be worth using a curve that is more discriminatory between 1 KB and 100 KB. We will see if the experimental results indicate this to be useful.

Priority Due to Importance

A transmission can have importance based on its content. UCCM obviously can't inspect the contents of each TR but can make heuristic judgments based on the types of TRs. Another aspect of importance relates to scarcity. If you miss one routine AIS report for a vessel, another report will be along in a few seconds, and it is very unlikely to convey radically new information. Any individual report is not likely to be that important. An image is different. Each is unique and you can't expect another to replace it without taking pains to go back to the same position and sight line to retake the image. It is also possible to argue that one color image can contain more useful data than one ESM or AIS report.

It is not that ESM or AIS reports are punished for being small and frequent. It is more a statistical idea that while any individual report is not that important, the collection of readings adds to an interpretation, is data that was requested, and therefore a sufficient number of them needs to be delivered. The importance score comes from a lookup table such as Table 4.

TR type	Score	Description
AIS	30	There are lots of them and they are often repeated.
ESM	40	There are lots of them, often repeated.
Radar return	40	Perhaps fewer than AIS, but can be repeated.
Track	60	Has more content than a radar return.
Thumbnail	90	Should be the precursor to any full image.
HRR image	100	Small but only asked for if really needed.
SAR image	70	Images carry more information than smaller TRs.
ISAR image	70	Same.
IR image	70	Same.
Visual image	80	Color contains more information than grayscale.

Table 4: Priority Due to Importance

There is an adjustment to importance that recognizes the practice of bundling. A bundle containing 1,000 readings is more important than a bundle containing one or two readings. A small factor (currently 0.1) is added for each reading in the bundle. The priority by importance factor is still constrained to grow no larger than 100.

Priority Due to Age

The general principle is that data gets stale with age. However, it is useful to divide TRs into three types for aging purposes. The priority due to age declines as the TR sits in the queue. When the priority goes under maxPriorityToKeep, the TR is deleted.

The curve needs to made good distinctions in the middle of its range, and less so for the extrema. We selected one sigmoid curve, but with different parameters for different situations. The generic curve is as follows:

$$\text{Priority} = (\mathit{scale} * e^{-((age-maxPt)^2 / sharpness^2)}) - \mathit{offset}$$

Where:

The *age* of the TR is given in minutes since it was added to the system.

scale determines the maximum values for the curve.

maxPt is the age for which the function is at maximum,

sharpness is a width factor. Small values produce sharper peaks around *age = maxPt*; large values broaden the sides.

offset: since exp() only returns positive numbers, *offset* is subtracted from all values so that the priority calculation can return values less than zero.

Each image is an individual, a snapshot in time. This data might be intended to guide the operator or data customer in the short term. "Where should I steer next? Is there a target I should focus on while I am here?" Then its value declines with passing time. Or the image might be intended for later detailed analysis, outside the mission time. In those cases, age does not matter, but then there is also no urgency to download the image quickly.

Images do not go stale as fast as other data. They are big, and the operators know they will take time to transmit. Each image is also unique. It may be possible to get another shot similar to any given image, but it requires navigating the aircraft to the same location, pointing the sensor in the same direction from the same altitude, and hoping the lighting conditions are similar. It is easier to give the operator every opportunity to get the original image.

The curve for imagery, shown in Figure 5, enforces a gradual decline in priority. The *scale* factor is 180. The maximum priority contribution comes at 10 minutes, and the *sharpness* factor is 20 to have a moderately broad maximum. The *offset* is 100.

The reasoning behind these values is that an image is given almost an hour to get into the radio system. Other factors, mainly its size, will decrease its total priority. The age factor does not start out at 100, to ensure that large images don't immediately take over the radio when they are put on the queue—smaller TRs get a chance to go. The age factor goes to 100 within 10 minutes. The age contribution goes to zero in about 30 minutes. If the image has not transmitted after 60 minutes or more, its utility is likely to be slender, and its age contribution to priority reflects that. Its age contribution will eventually go to -100, ensuring the image will be subject to garbage collection. See the section below on thumbnail mode for a way operators can ensure that they get the images they need, and quickly.

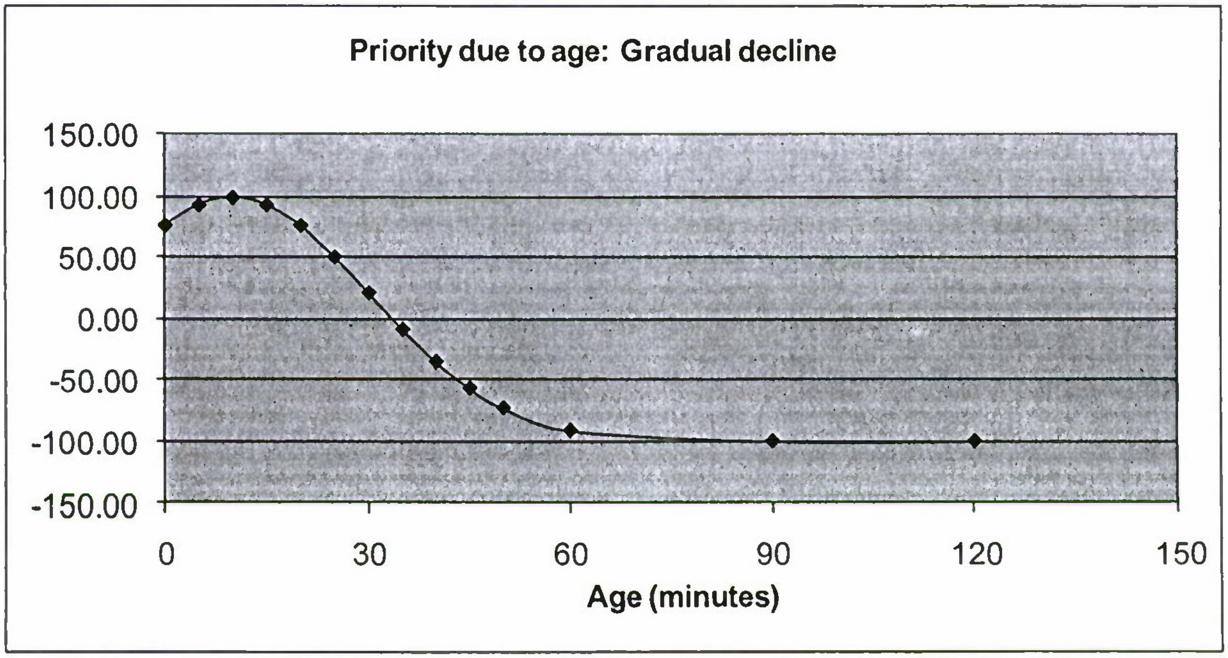


Figure 5: Priority Due to Age of TRs with Slow Decline

AIS and ESM readings age much more quickly than images. We assume that if the aircraft is actively monitoring an area, a track still on the queue after about 15 minutes has either been superseded, or the aircraft is now doing some other task. We don't age AIS messages at the fastest rate because older messages can be useful to provide missing points on a track. Knowing where the ship is now is the most important, but it can be useful to know

where it has been. ESM ages at the medium rate because it is hard to know what a reading represents. Without knowing who or what emitted the signal, it is hard to assume one reading will be soon replaced by an equivalent one. And some emitters are on only for brief periods. The "quick death" curve for age takes the age factor to zero within about 10 minutes and to -100 in about 20 minutes. Age is only about a third of the total priority score, so often even this is not enough to force the TR below the threshold. The quick-death curve flattens out at -130 after half an hour on the queue. This curve is shown in Figure 6. The *scale* parameter for this curve is 230. The *maxPt* is set for one minute and the *sharpness* is 8. The *offset* is 130.

We regard this solution as useful but less than elegant. Each priority factor is supposed to range from -100 to 100. Allowing a factor to go to -130 is implicitly changing the relative weighting of the factors in certain situations. We did it to get the numbers to come out right, but will look for a better way to accomplish the same end. An alternative might be to have rules explicitly decide to kill specific TRs. We may end up doing that, but for now will try to do it just with priority.

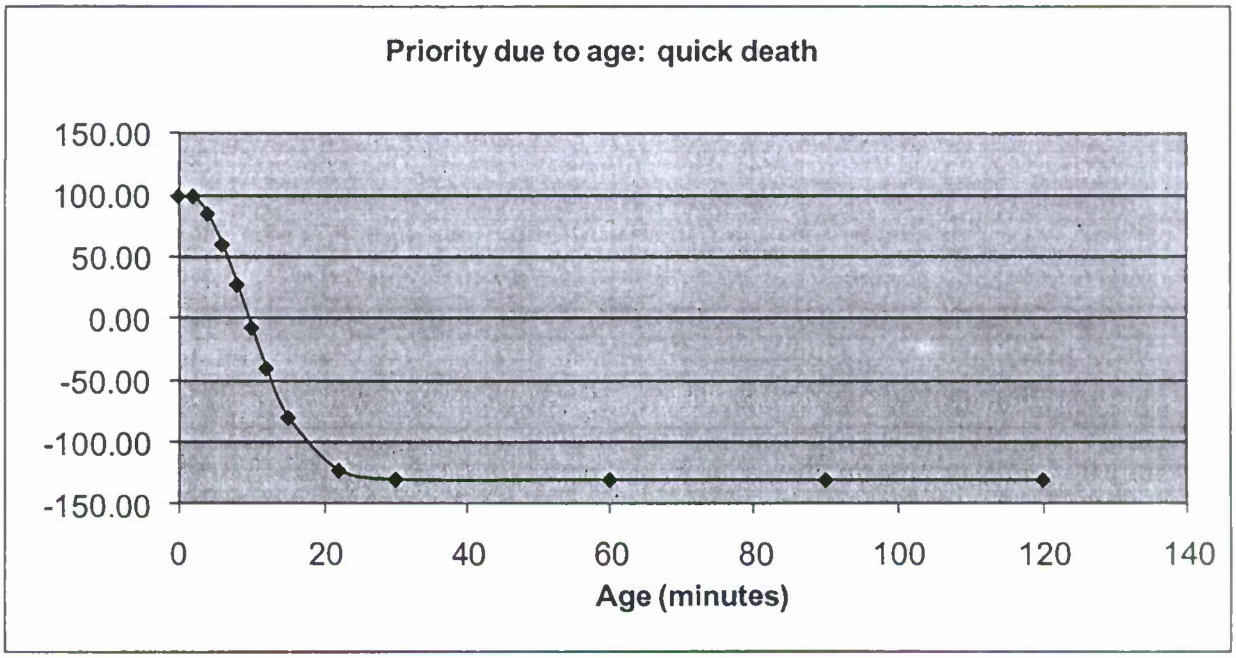


Figure 6: Priority Due To Age for TRs with Moderate Lifetime

Tracks and radar returns are treated more harshly, as the expectation is that a reading will be replaced within minutes or less if the aircraft is actively monitoring an area. The "galloping death" curve of Figure 7 is used to age these out of the system quickly. This curve gives the sensor reading a chance to get sent immediately due to its small size. The age factor goes to zero in about 5 minutes and to -100 in 10 minutes. It flattens out at -130. The *scale* parameter for this curve is 230 with an *offset* 130. The *maxPt* is at 1 minute and *sharpness* is 4.

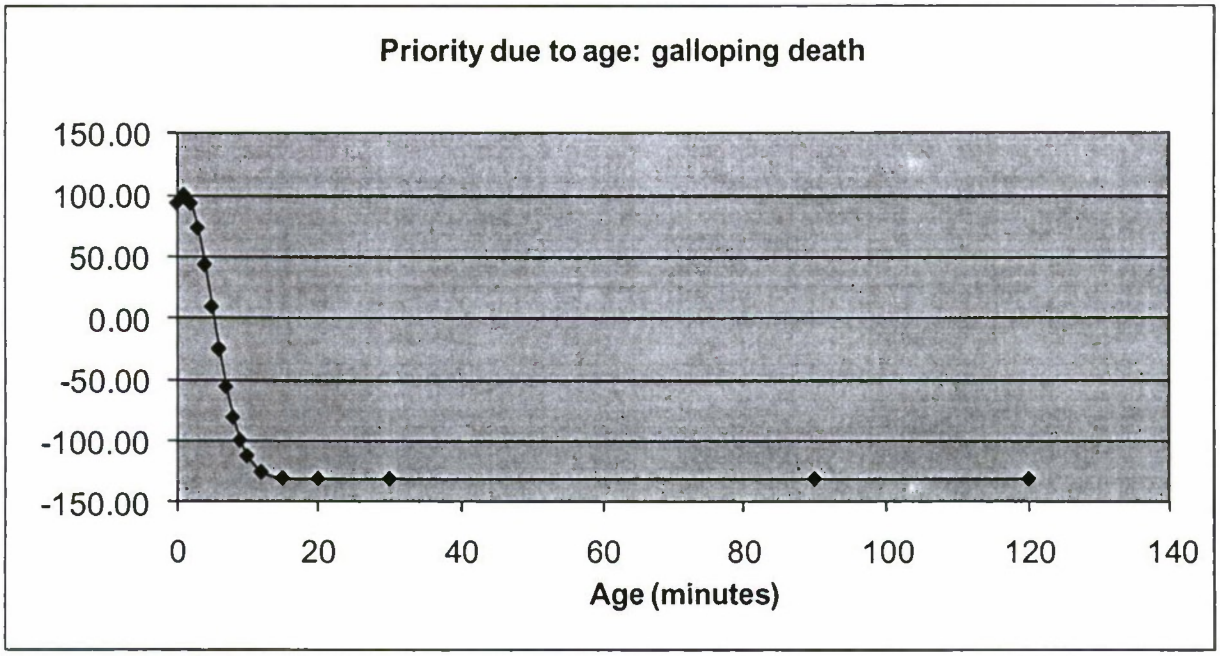


Figure 7: Priority Due to Age for TRs With Minimal Lifetime

Time inexorably advances, so age must be a dynamic factor. The queue component throws queued TRs back into the rule system on a scheduled basis to have their priority due to age re-evaluated. The reprioritization should occur at least every two or three minutes given the short lifetime of many types of TRs. It also needs to be done frequently, so that newly arrived TRs really do have an advantage over older TRs that will otherwise still carry their priority from age = 0.

Priority Due to Operator Intent

The clearest indication of a TR's value to the operator is when the operator declares their intent. If the operator takes an action that forces the TR to be downloaded now, when it would not have been otherwise, that speaks for strong perceived value. Other actions declare other types of intent.

A transmission request has an intent field that is separate from its type or status. This is mainly set from operator commands and from defaults. Values include Selected, Held, Thumbnail, and Not specified (the default). There is also a degree of intent in the type of the TR, specifically HRR images, radar returns, and thumbnails. We capture that in the importance factor of the prioritization because that is a static decision.

The default values of intent come from a lookup. The current values are given in Table 5. Thumbnails, HRR images, and radar returns have an implicit intent based on what they are. An operator won't ask for HRR or radar returns unless they specifically want them. And if they do specify that mode, the assumption is that they want the data now. Thumbnails have high intent because they must be transmitted quickly to give the operator a chance to specify intent with respect to the underlying image.

Intent	Score	Description
Not specified	40	Default value.
Held	-40	Low value inhibits transmission.
Selected	100	Very clear signal of intent.
Thumbnail	75	Selected based on contentType, not intent.
HRR image	90	Selected based on contentType, not intent.
Radar return	85	Selected based on contentType, not intent.
Deleted	-20	Notifies FIFO queue to update metrics .

Table 5: Priority Due to Intent

A thumbnail can be sent, and the operator specifies that the image should be deleted. The rules will delete the image, but there will already be a FIFO copy of the image. Queueing that image on the FIFO queue again with intent = deleted tells the queue to adjust the stored priority downward in light of the new information about the operator's intent. Similarly, queueing a TR to the FIFO queue with intent = selected will adjust the stored priority upward. Priorities are not used for selection in the FIFO queue; these adjustments are done to make the metrics based on priority more accurate.

An operator can declare one sensor per channel as generally selected (selectedType). As long as that policy is active, all new readings from that sensor become TRs with intent = selected. This is a way for the operator to state "I am specifically dealing with this sensor right now; give it some preference." The policy can be applied to any sensor. AIS, for instance, typically gets only a middling priority due to a low Importance score. This is a way to give AIS in general a boost so that more of them get transmitted compared to other types. An operator can similarly declare one content type per channel to be held (heldType).

Running in Thumbnail Mode

Since imagery is so much larger than what the other sensors produce, the operator can set UCCM into "thumbnail mode." As a new image arrives, a small thumbnail version is created. The thumbnail TR carries the size, arrival time, and content type of the full image, but its content is the thumbnail image, and the full image is only linked to. Thumbnails compete for radio time like all other TRs. When the operator selects a thumbnail for download, the full-sized image taken out of a buffer and queued to be sent. The image gets priority boosts from having Intent = selected, and from its age starting from the time of selection, not from when the image arrived in the system.

Thumbnails serve both as a way to explicitly collect the operator's intent and to provide means to select and weed out images at minimal bandwidth cost. The operator's interface will have a special page for viewing thumbnails. The operator can select any thumbnail and make a choice:

- **Mark as selected.** The intent is "send the full image now; I want it." If possible, a facility that allows the operator to marquee only the part of the image they actually want will be provided. The TR for the thumbnail is replaced with a new TR for the full image. Its Intent score is set to 100, and its age starts at zero to give it two boosts in priority.

- **Hold**. The semantics are that the image is useful, it needs to be sent at or before the end of the mission, but the operator doesn't need it immediately. Operators will immediately download images that help them or their customers make real-time decisions. They will "hold" images useful for offline analysis. A held TR is never sent unless all active TRs are sent. It is immune from aging and from priority-based thresholds.
- **Delete**. This is not a useful image. Delete it on the BAMS and reclaim its memory.
- **Ignore**. If the operator does nothing, the image remains in the image buffer.

A TR that is a thumbnail gets an Intent score of 75. It isn't explicitly selected by the operator but the design intent is that it has a higher than normal chance of being sent.

Weighting Factors

Prioritization is based on size, age, intent, and importance. Importance is the least of the factors. It reflects only a static judgment of the contentType. Size is the next most important factor. The various signals sensors are very close in size, and should not be judged on that basis anyway. The size factor mainly serves to distinguish between signals, images, and thumbnails. It also encodes the thought that large TRs, which hang up many others while being transmitted, are less preferred than smaller TRs. Age has importance because it is a dynamic factor that encodes the shelf-life theory. Two otherwise identical TRs can have different histories due to when they were collected, and what else was going on in the system. The operator's intent, when it is known or can be inferred, is the most important factor, so it should have the largest weighting.

Stated in this way, the system suffers from "banding." Each TR has a small range of priorities for age = 0 that depends on the size variation of those types. However, if all AIS readings have an initial priority of 70.10 or more, and all ESMs have an initial priority of 70.09 or less, then no ESM will be sent until all AISs are sent. The decision to send is based on spurious precision. To get around this, a fuzz mode is available and is on by default. In this mode, 2% of the score is random. Prioritization should not be taken as an exact science, and this breaks up the bands of TRs by type since all the of small TRs start out with age = 0 priorities within a point or two of 70. The final set of weightings is shown in Table 6.

Factor	Weight
Size	25%
Age	29%
Intent	34%
Importance	10%
Fuzz	2%

Table 6: Priority Factor Weightings

Reprioritization

UCCM actively manages the PTRs on a queue; the initial priority is not final. There are several scenarios in which some or all of the PTRs can be reprioritized:

- Age is an important factor. The priority due to age will be recalculated regularly to ensure that newer items are processed and so that items have a chance to die of old

age. Images might be reassessed every ten minutes or so. Smaller TRs should be recalculated more often, considering how fast their curves decline.

- It is possible for UCCM to use a customized prioritization for a specific channel. A channel containing nothing but AIS and ESM might use a different curve for priority by size than one for images, to be more discriminating. Changes in channel assignments of sensors may trigger reprioritization of the channel.
- If a channel becomes unavailable, UCCM may decide to distribute its PTRs to other queues. When this happens, the recipient channel(s) typically need to be reprioritized in case of specialized priority schemes.
- Major changes in operating conditions could trigger reprioritization. Potentially, a command from the operator could trigger it.

When a TR is reprioritized, its content, content type, and arrival time are preserved. The main priority field is cleared, along with any of the subfields that need to be recalculated. The rules automatically pick up from the correct place in prioritization pipeline.

The current scheme is to ensure all PTRs are reprioritized within a long interval. Instead of all being dumped into the rule system at once, they are divided into cohorts within the long interval. When a number of PTRs are sent back to the rule system, they are injected with a short delay between each so that the reprioritization work is intermixed with the regular prioritization of newly arrived TRs. These are all settable parameters.

Transmission Request Life History

When a transmission request arrives at UCCM, it is prioritized with age = 0. There won't have been time for a thumbnail to be created from an image, sent, and then selected, but the operator might have specified that all of this sensor's output be held or selected. Absent anything else, thumbnails will go onto the queue instead of the full images. All the signals TRs will start with about the same priority, which is less than the generic thumbnail priority. If images do go directly onto the queue, they will have priorities lower than the other TRs due to their size.

All things being equal and if bandwidth is good, then the radio will send the few new thumbnails, clear out recent small TRs, and then send an image. Things aren't always equal, and bandwidth can be disadvantageous. In this case, remember that there are three channels, and UCCM will not always have to let all types of TRs compete on the same channel. Also remember that TRs do not age at the same rate, and that operator selection can have major impacts.

When over an hour into a mission, we often expect to see a cluster of TRs held for later. These TRs are valuable data, but taken out of the competition. If conditions are adverse, we will start to see some of the small (but numerous) TRs age below the threshold and get deleted. By definition, these are older readings that have been superseded, and/or have declining relevance to the operator's current concerns.

There is a limbo when a TR has a priority less than minPriorityToSend, but above minPriorityToKeep. These TRs will not be sent until their priority increases. In other domains, such as helicopter communications, there are a number of dynamic factors that can accomplish this. In the BAMS domain, only operator selection can boost TR priority. Otherwise their

priority continues to decline with age until it goes below the second threshold, and the TR is garbage collected.

The Operator's Interface

The goal of this SBIR was to create a prioritization method and show its utility for BAMS. We did not take the design of the operator's user interface as part of our scope, but we did a simple mock-up anyway, to show how prioritization would look to the operator.

The first screen, Figure 8, allows the operator to assign sensors to channels. It should also reflect current reality. If the rules are extended to reassign sensors under certain circumstances, this screen should reflect those new assignments.

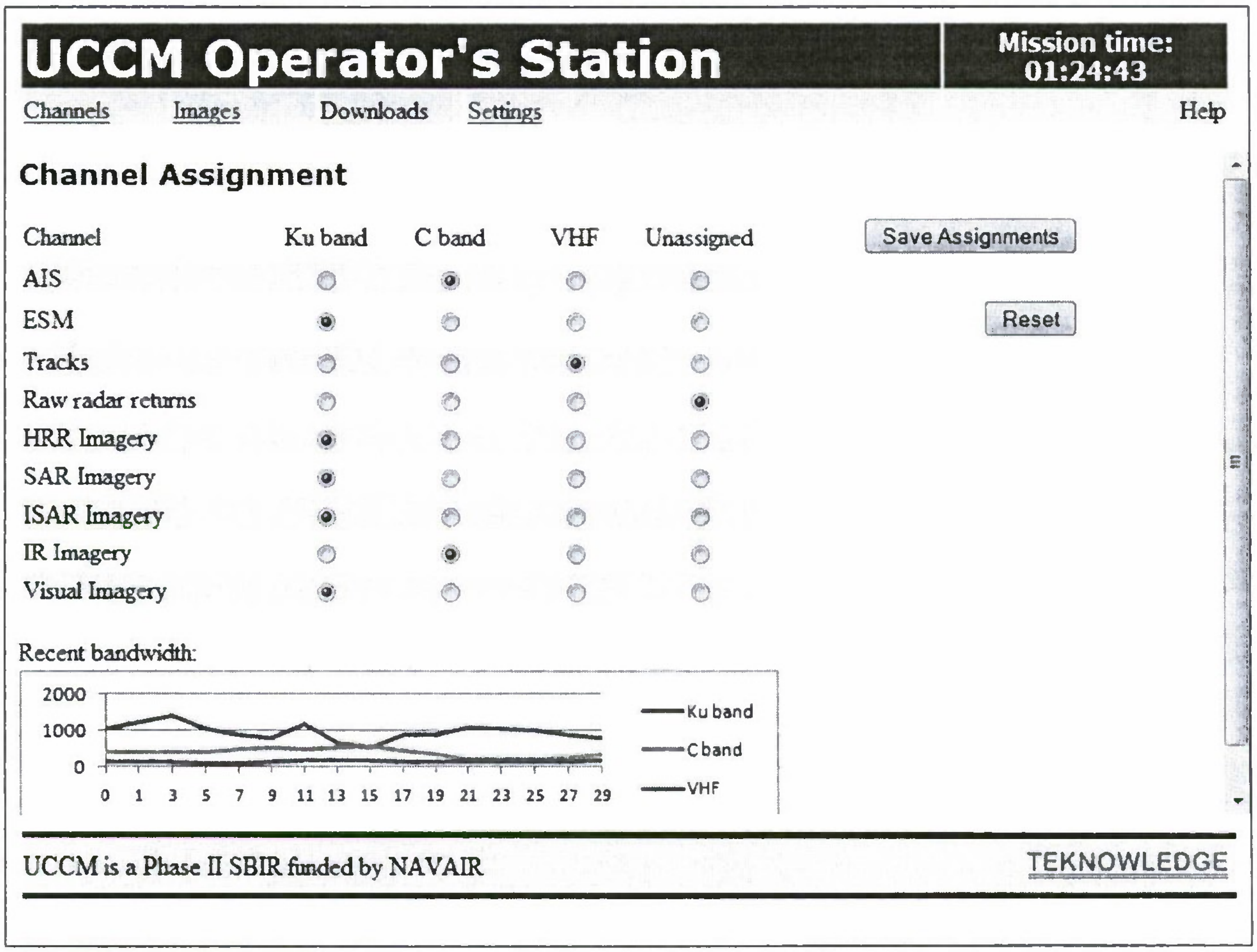


Figure 8: The Operator's Channel Selection Screen

The image-selection screen shown in Figure 9 is one of the main screens. This uses a familiar slide-sorter layout to display the thumbnails that have downloaded so far. The operator can quickly see what each image is, how big it is, and its age (which should be refreshed on some periodic basis). Each image is accompanied by a set of operations the operator can take. The delete option causes the full image on the vehicle to be deleted from the image buffer, and the thumbnail will be deleted in this display. If the action is to hold the image, UCCM will queue the image with intent = Hold, as described above. When the action is to download, the full

image will be put in the queue to be sent, with intent marked as selected so it gets some boost in priority.

The advantage of the image-selection protocol is that no image is sent unless asked for. Since around half of visual images have low usability due to clouds and other poor conditions, this is an easy way to weed them out. In some cases, just seeing the snapshots might tell the operator what they need to know, if the image was not taken for analysis purposes. As a refinement, the interface should offer a marquee facility. The third image shows this. Everything outside the yellow marquee is masked out. When the operator selects this image to download, only the selected portion of the image is sent, further reducing bandwidth.

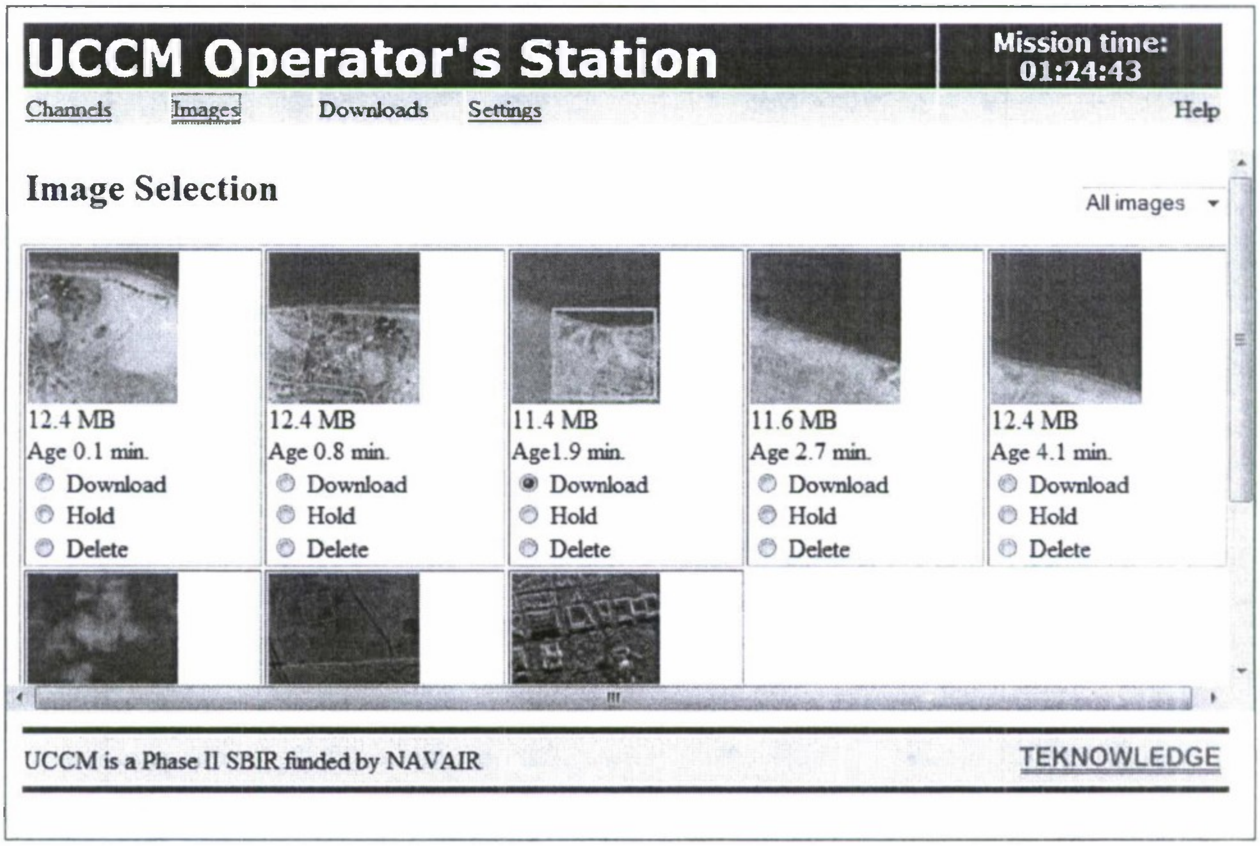


Figure 9: The Image Selection Screen

The final UCCM screen is the settings screen shown in Figure 10. Each channel has its own settings for which sensor type is held or selected (if any). Each channel also has a small sparklines display for recent bandwidth, as context for any decisions to change settings. Setting selectedType = Visual image would be disastrous unless bandwidth has been high recently.

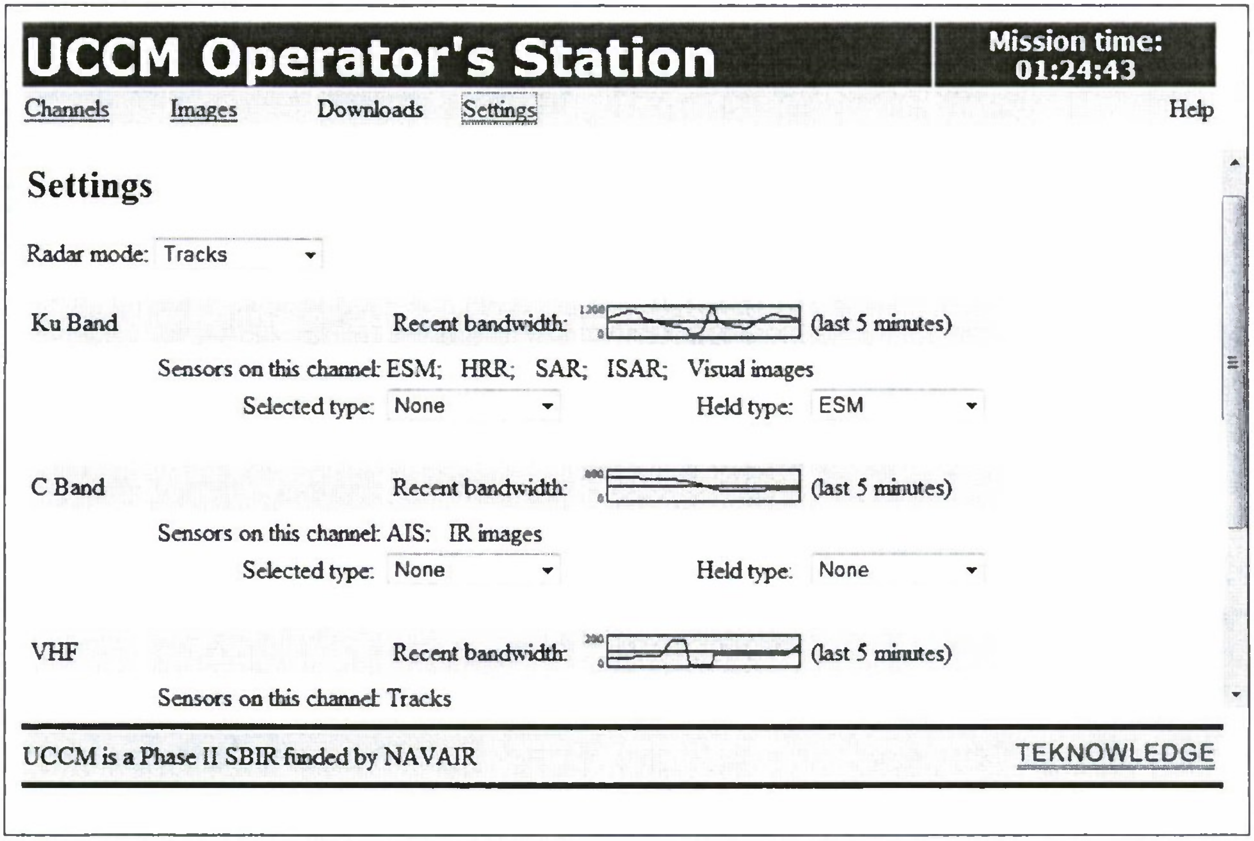


Figure 10: Selecting Channel Settings

Simulation Experiments

The simulated aircraft is defined by a data generator and a simulated radio system. The ActionWeb test harness is used for the data generation. This test harness reads a set of scenario specs and uses them to generate a sequence of TRs at runtime. There are three XML files that specify a scenario:

1. The metadata file defines the events to be generated and their fields.
2. The metadataGeneration file defines the frequency for each event, and defines statistical distributions by which their fields take on values. Events are generated according to a Poisson distribution characterized by a *lambda* parameter, which roughly corresponds to one event every *lambda* seconds. Each field may be generated as a constant, a random linear distribution between N_1 and N_2, a parametrized Gaussian distribution, or a user-provided distribution. Field values can also be set by simple calculations of other fields.
3. The scenario file provides specific events along a timeline. In this way, the scenario can specify bandwidth and operating changes, and cause specific events to happen at specific times.

The test harness runs from a simulated clock. On a standard laptop with a moderate TR arrival rate, we are able to run between 1.5 and 3.0 times wall-clock speed. There is a UCCM radio simulation and one for the FIFO radio. Both simulations pull the top item off their respective

queues, read the size, delay until the TR is "sent" according to current bandwidth, and then repeat.

We present here several illustrative test runs that highlight particular UCCM features. More details of these runs, with annotated snapshots, are in the appendices.

- Case I has bandwidth roughly equivalent to the rate at which the sensors are outputting MB. The presence of a small number of images in the mix forces the FIFO side to queue up and delay TRs that are increasingly out dated by the time they are sent.
- Case II features a more adverse bandwidth situation. An unprioritized radio cannot manage its resources. Its queue grows according to classic queuing theory until the system can't handle it. After a while, it is sending TRs that are 30–60 minutes old. The UCCM radio is able to intelligently drop the less valuable TRs and manage its resources.
- Case III shows how UCCM handles special requests. Bandwidth is high enough that everything will be sent with no more than 5–10 minutes of delay. However, UCCM is able to move HRR images, radar returns from scanning mode, and specially requested tracks to the front of the queue.

Not shown is the control case. UCCM will perform identically to FIFO when bandwidth is comfortably greater than the sensor output rate, there are no special requests from the operator, and there are no images in the output stream for this channel.

UCCM Results

UCCM uses one simple and consistent mechanism—calculated priorities—to achieve a wide variety of useful behaviors. The following list discusses the main communications benefits UCCM offers.

- **Efficient use of bandwidth for images.** UCCM's thumbnail mode ensures that the only images that transmit are images the operator specifically wants. No bandwidth is wasted on images of fog banks or empty ocean. In some cases, the thumbnail itself might be sufficient. The protocol of thumbnail followed by selection not only optimizes the transmission of images, but ensures they don't unduly impact other transmissions.
- **UCCM sends newer data in preference to old.** It sends almost all TRs soon after receiving them, on the theory that newly arrived data about some entity generally supersedes older data about the same entity. Use of priorities means it doesn't have to check a TR's identity to achieve this end. The older data gets sent if bandwidth permits.

  The unprioritized radio does not use priorities, and sends data in the order received. Then each large image delays all TRs after it, reducing their timely value. In a long mission when bandwidth is often low, the difference between the two communication regimes can be quite dramatic.
- **UCCM is able to dynamically manage its own resources.** UCCM is able to delete TRs, hold them for later, and adapt its own operating characteristics. This means it can maintain a much more reliable and consistent operating profile. The unprioritized regime, in contrast, is subject to classic runaway queue problems when arrivals exceed departures. Since the UCCM queue deletes TRs when they get too old instead of sending them, the average priority for UCCM tends to stay very constant.

- **A sensor reading may have significant value without needing to take up bandwidth now.** Perhaps it is data requested by some other group, who will analyze it later. UCCM is able to hold such data until bandwidth becomes available or until the end of the mission. Either way, the mechanism frees up bandwidth for data that is more immediately needed.
- **UCCM is able to honor specific requests from operators.** Specific images may be deleted, downloaded, or held. Similarly, the operator can designate entire runs of one sensor's readings to be held or to be specially selected. It can also build in expert operator knowledge so that it does not need to be told everything. It knows that HRR images and radar returns are uncommon requests that should get expedited treatment so they retain their real-time utility.
- **UCCM prioritization and other functionality is highly parametric.** There are many levers that enable developers to adapt UCCM to application needs, and for operators to take as much control as they need. We have defined a small number of settings that operators can use, instead of overwhelming them with an excessive number of choices.

Performance and Scalability

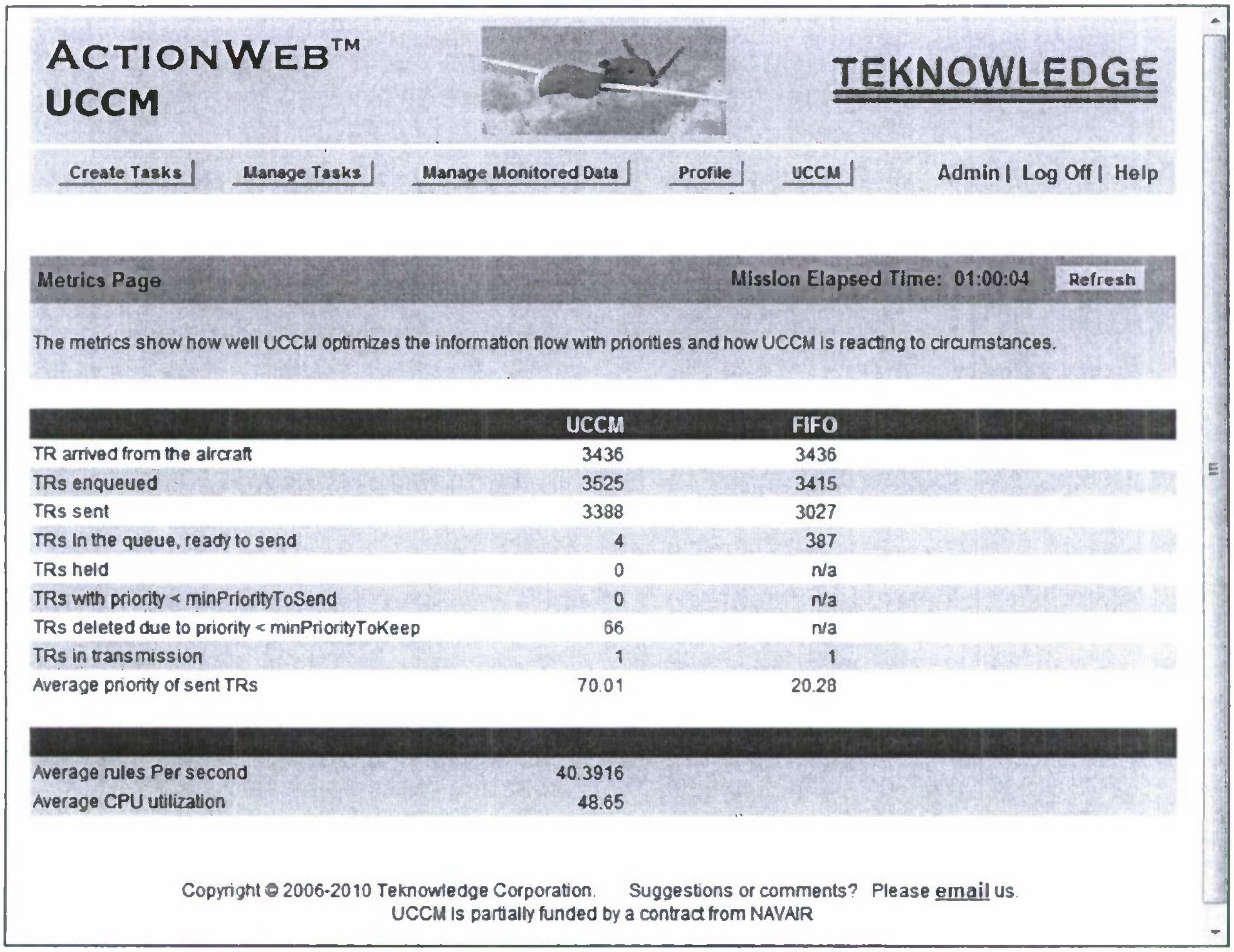
Phase II work tested the UCCM system combined with the comparative FIFO simulation. A deployed system will drop the FIFO simulation, leading to better performance using fewer resources. Benchmarking of a UCCM-only system has been left for a possible Phase III.

Metrics of Performance

We have a number of performance metrics to guide our systems work. The rough measure of how hard the system is working is given by measuring rule executions per second, and pairing that with the percentage of the CPU being used. We try to quote all rules/sec figures at 50% processor utilization. We also check the percent of CPU in the Windows Resource Manager, and the process size in the Task Manager.

Internal performance depends on two components. TRs go first through the rules engine. The main issues here are how fast the rules are processing, and the number of TRs in the rules engine's working memory. TRs enter the rules system, are prioritized and enqueued to the priority queue, and are deleted from both components when the TR is sent. The rules engine alternates between filling a rules agenda with rules enabled to fire by events in the working memory, and executing rules from this agenda. The length of this agenda is another guide to how hard the rules are working.

The UCCM developer's interface provides a number of screens that display details of the UCCM queue and the FIFO queue. The overall summary is provided by the screen shown in Figure 11. This page allows us to trace the overall process by steps. "TRs arrived from the aircraft" counts TRs from the test harness (or the sensors they simulate). "TRs enqueued" counts those that actually get put on the priority queue. Before that point, some may be deleted by the operator, some images may be held in an image buffer, and the rules may create new TRs such as thumbnails. These factors explain why the UCCM total is not the same as the FIFO total.



	UCCM	FIFO
TR arrived from the aircraft	3436	3436
TRs enqueued	3525	3415
TRs sent	3388	3027
TRs in the queue, ready to send	4	387
TRs held	0	n/a
TRs with priority < minPriorityToSend	0	n/a
TRs deleted due to priority < minPriorityToKeep	66	n/a
TRs in transmission	1	1
Average priority of sent TRs	70.01	20.28

Average rules Per second	40.3916
Average CPU utilization	48.65

Figure 11: The UCCM Metrics Page

Once in the priority queue, "TRs sent" counts the number of TRs that are transmitted by the radio. " TRs in the queue" counts the active TRs and is a key factor. These are the main source of load on this component. These active TRs are subject to reprioritization and may also be transmitted. The next line counts TRs that are actively in the queue, but in held status. These are also subject to reprioritization. The next two lines count TRs in limbo due to their priority, and TRs that have been deleted for low priority. There is always either zero or one TR in transmission. The number enqueued should equal the sum of TRs between TRs sent and TRs in transmission.

Performance Analysis

The UCCM standard configuration uses Java 5, Apache Tomcat 6, and the H2 embedded SQL database. All performance figures cited in this report were taken on a 64-bit Windows 7 laptop with a 2.5 GHz dual core processor and 4 GB of RAM. We typically use either the Firefox browser or the Iron browser from SRWare. Iron is based on the Google Chrome open-source browser, but strips out Google's user tracking and spyware. We use it for testing because it has a very small footprint and because it handles our AJAX-based charts better than the Microsoft Internet Explorer or the Firefox browsers. All components run on the same standard laptop.

In this configuration, UCCM runs at 40 rules/sec. at 50% CPU utilization. We estimate that it will take 80–100 rules/sec. to handle the worst-case scenarios discussed above. We are confident we will be able to improve the system to that level. First, a deployed system will not run the FIFO simulation. That alone can cut the work in half. Second, we have not run out of bottlenecks to identify and optimize. During Phase II we increased speed by about 140 times, mostly by database optimizations. The next step on that path, left for Phase III, would be to redesign the database schema for the storage of events. The current ActionWeb schema is a legacy design optimized for an earlier data representation. Since event I/O is in several central loops, a more efficient schema can cut the number of queries significantly, and make a dramatic difference in performance.

In long-duration runs, we have seen UCCM grow to consume as much as 65% of the standard machine's CPU. Eclipse/Tomcat ran at 5–10%. The Iron browser stays at a few percent if you don't visit one of the two AJAX metrics charts. These charts grow to thousands of points and are updated in real time. Most browsers just can't handle that after a few thousand points. Iron seems to be able to throttle itself back, but still consumes a lot of cycles to compute the updates. Since the database is embedded, it is counted as part of the UCCM process. A deployed version of UCCM would not run the FIFO simulation, would not be running a browser, and would run a Web server directly instead of through the Eclipse environment we run for development. It would also not run the server-push code to update metrics charts.

UCCM to date has been a CPU-bound process, but we have identified and solved several major space issues. We have not spent much time yet to minimize the image size. We currently compile the system to start with 128 MB of heap, and allow it to grow to some large size like 1,024 or 1,536 MB. We have not hit that limit yet. While TRs are created, processed, and deleted, the garbage collector does not bring the image size down to starting size. It still grows over time. We recently realized this is due to several accumulative DB stores that are now part of the image since we adopted an embedded DB. One is the ActionWeb Incident History that logs the rules fired for each TR, operator action, and several other processes. This is a developer's tool and should be turned off in a deployed version. The other major store is the metrics data, which accumulates data on TRs, even after they are deleted. This should also be turned off for deployment, or periodically written to a file instead of retained in the image. These two changes should hold the image size to what is really needed to run the system. Our educated guess is that this would be 200–400 MB depending on the number of active TRs, and absent optimization of image size.

When bandwidth is poor and arrival rate is high, more TRs build up within the system. The FIFO model has no way to deal with these, and they accumulate without limit. UCCM has already shown the ability to keep its own queues manageable. A possible Phase III task would be to add a few rules to detect when the system is bumping into performance thresholds and to take corrective action. A few examples:

- If the queue is too long because data is arriving too fast, UCCM can ask the sensors to accumulate larger bundles of readings.
- If the queue grows because this channel has terrible bandwidth, consider moving one of the sensors to a better channel.
- In either case, the system can choose to switch to a different age curve that kills off older TRs faster, or can increase the thresholds for where to purge TRs.

- If reprioritizations are taking too much of the interval allotted to them, the system can switch to a less frequent schedule until conditions improve.

The other major issue we have identified are the resources devoted to threads. ActionWeb uses a large number of threads in the rule-execution phase. We reduced the storage allocated to new threads to 128 KB, and could possibly reduce it further. This helps considerably, but there are still corner cases in which the number of threads could be an issue. We believe one architectural change in the underlying ActionWeb platform will solve the problem. ActionWeb creates a new thread for every action. Most ActionWeb applications to date have lower data rates than UCCM, and a high proportion of actions are externally focused. If the action operates machinery, invokes a Web service, or interacts with humans, the lengthy time frame mandates a separate thread. UCCM processes a lot of events (although bundling holds that in check) and at least 90% of the rule actions are internal—they only modify working memory. We would add a switch to make all internal actions use the same thread. This would cut the maximum number of live threads from hundreds to dozens.

In summary, we have identified and solved several performance challenges. We now have high confidence that UCCM can be extended to meet foreseeable space and processing limits because we have already analyzed the steps to take. UCCM should scale to handle the BAMS application.

UCCM's Maturity Level

UCCM is at TRL (Technical Readiness Level) 4. TRL 4 requires the basic components of the system to be integrated into a running system that runs in a low-fidelity laboratory environment. We have had that since early in Phase II, over a year ago. UCCM needs to be integrated into a larger system including a radio subsystem and sensor inputs. These would be BAMS simulations instead of our own simulations, and the APIs would be different, but these are minor differences as far as the function of UCCM is concerned. It will be integrated as a black box that needs very little from the surrounding system.

UCCM is at SRL (Software Readiness Level) 4. It is a stand-alone system that solves representative data sets. All the components of UCCM clearly work together. There is more systems work to do, as outlined above, but the state of the work is definitely better than the "relatively primitive efficiency and robustness" in the description of level 4. The software has been under configuration management since the start of Phase I, and we feel this should be required much earlier than TRL 5. COTS/GOTS components in the UCCM architecture have been identified.

The software is more mature than a typical SBIR because UCCM is an application of the ActionWeb platform. Teknowledge has made significant investment into ActionWeb outside of UCCM, and intends to sell ActionWeb as a product and/or use it as the basis of other application projects.

Summary of Phase II Work

The BAMS UAS, when fully operational, will provide a leap in the amount and quality of intelligence available to the warfighter. BAMS communications will almost certainly be overwhelmed by the amount of data its sensors can produce. Not all the readings have equal value. In fact, identifying which readings have the most value at any time is a challenging problem because the bandwidth can change without notice, the needs of the operators change as the mission evolves, and most other aspects of the mission are dynamic. UCCM provides a simple, uniform framework based on heuristic priority calculations that adapt the communication flow to match the changing situation. As shown in the experimental results, a few simple mechanisms interact with the dynamic environment to create complex behavior that optimizes the flow of data.

UCCM solves the problem of providing the right data, to the right people, at the time they need it. The prioritization algorithm handles the multifactor tradeoff that is required. The solution is not perfect, but when the relevant parties agree on the heuristics, it should be close. The experimental work clearly shows that the lack of any prioritization scheme leads to a very poor solution of the communication problem. The main issue is not whether to prioritize, but the exact details of the prioritization and how it is used.

UCCM is not a point solution for the BAMS application. It is a general strategy for optimizing the flow of valuable data from many producers. This platform is specialized for the BAMS problem by defining the specific metadata available with each TR, creating the connectors that obtain this metadata, and providing the exact rules for computing a priority in this problem domain. We have already done illustrative work in the helicopter domain, where many independent applications send data at a small crew. We have also considered UCCM in the context of a submarine that surfaces and needs to make best use of a limited communications window. We see no reason why UCCM should not be applicable to most situations involving mobile entities that need data communications in challenging environments.

The ultimate test is whether UCCM benefits the warfigher. It provides value in three areas:

1. UCCM increases BAMS' ability to deliver crucial intelligence to the warfighter, in a timeframe where it makes a difference in the battlespace.
2. As bandwidth fluctuates, UCCM ensures that the most important transmissions go out first.
3. UCCM automates some of the communications management so that the BAMS operators can spend more time developing the intelligence requested by warfighters.

A fourth benefit accrues more to the larger organization than to the individual warfighter. Military applications are often not written to be bandwidth aware or concerned with individuals' needs. BAMS is only a simple instance of this—each sensor just outputs readings regardless of whether they can be transmitted or whether anybody wants them. Inserting UCCM between applications and the radio injects knowledge of individual needs and local bandwidth without having to rewrite major legacy applications to do so.

Case I: Moderate Bandwidth

In Case I, all sensors output to the same channel. The fewer sensors on a channel, the less UCCM has to work. The bandwidth is calculated to be approximately the same as the output rate of the sensors, at a modest output rate. We ran this case for 60 minutes. The actual bandwidth profile during the first 30 minutes is shown in Figure 12. The bandwidth continued at a constant rate thereafter.

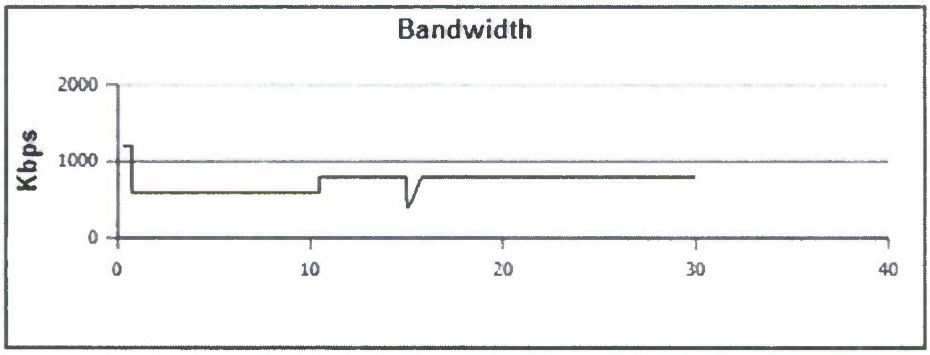
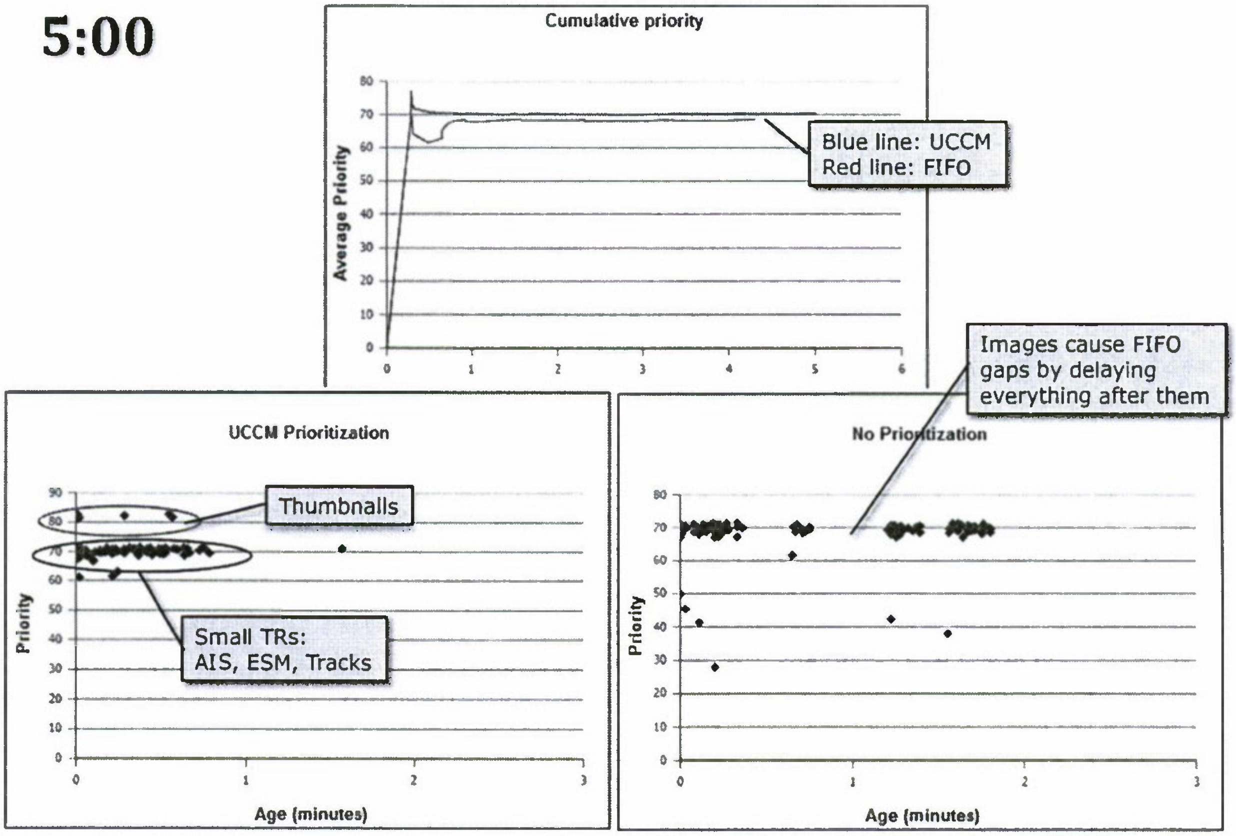


Figure 12: Case I Initial Bandwidth Profile

The channel sees a steady stream of small TRs (AIS, ESM, and tracks) with a few EO and IR images mixed in. There is a visual image about once a minute, and an IR image about once every 40 seconds. The test harness can simulate operator commands too, since they appear in UCCM as events like any other event. The operator selects and operates on an image approximately every 40 seconds. 40% of the operator actions select an image for download, 40% select the image to be deleted, and the rest are held. This case features only a few excursions from this steady background. A run of ESM is collected as the heldType in the first 10 minutes. Later in the mission, some SAR images are collected instead of tracks.

We will present all cases as a series of snapshots of metrics screens, typically at five minute intervals. This is a good interval to see what is happening without getting too verbose about it. It can be informative to flip through the pages of the case quickly as a simple animation.

5:00

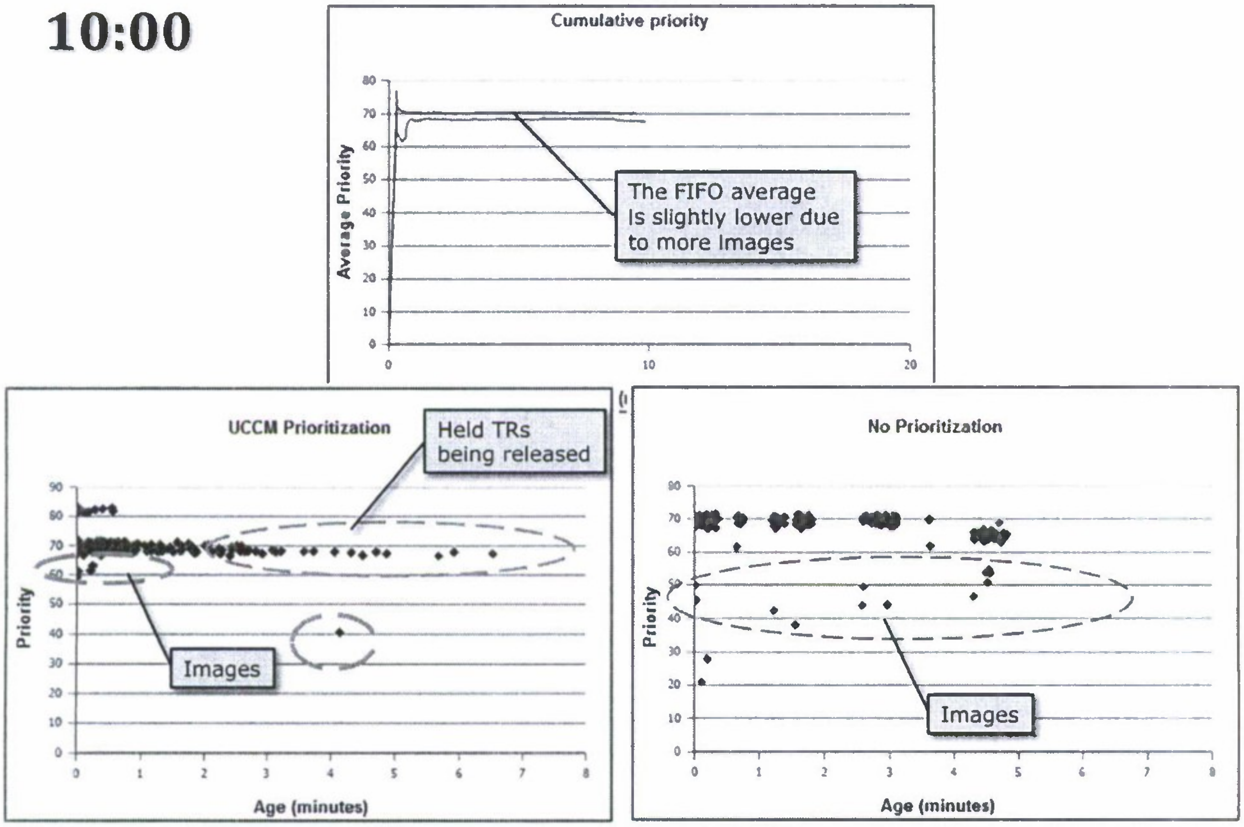


The upper screen of these screenshots shows the cumulative average priority of all TRs sent through a radio. The lower red line shows the unprioritized FIFO radio, while the upper blue line shows UCCM. There is generally some wobble in the early part of a run as the average contains only a few points, and one or two TRs can make a major difference.

The bottom pair of charts displays the priority of each TR versus its age. Even after only five minutes, both show common patterns. By far, the majority of TRs are either AIS, ESM, or tracks. At age close to zero, their starting priority is close to 70 for all three, with some variation due to the larger bundles of tracks and ESM readings. The cluster at 70 shows these TRs all being sent within a minute of being put on the queue. A small cluster of points around 80 are the thumbnails for the images. Thumbnails are small and have a boost in priority to make sure they are sent promptly, instead of the images they stand for. In the first five minutes, a couple of images are sent, represented by the points around 60.

The FIFO chart presents a different picture. All FIFO TRs are prioritized the same as UCCM TRs. The priority is for comparison and is not used to select TRs to be sent. The small TRs also start with priority distributed around 70. One difference is that the cluster is stretched out in age compared to UCCM. No thumbnails are used, therefore images are sent in the order they arrived. These are the points with priority of 60 or less. Each image takes a while to transmit, so introduces a gap in the band of small TRs. Note that each gap has a low-priority dot directly below. This is the image that caused the gap. The smaller the priority, the larger the image, and the longer it took to transmit.

10:00

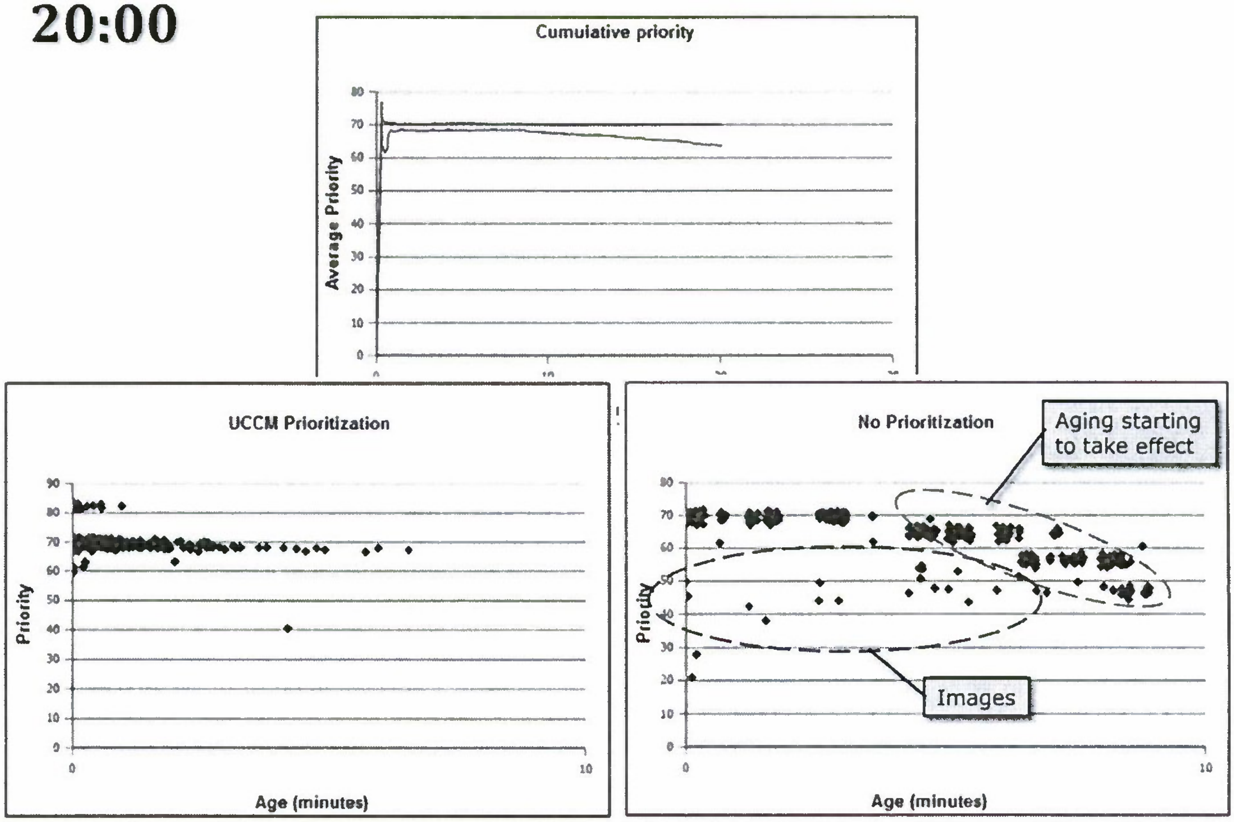


When the data stream is composed entirely of small TRs, and the bandwidth exceeds the sensor output rate, FIFO would have equivalent performance to UCCM. The FIFO average priority is slightly lower because more images are sent, with lower priority. The UCCM queue also sends a number of higher priority thumbnails.

The scatter charts continue their previous pattern. UCCM sends most of its TRs quickly. Bandwidth is good, so the run of held ESMs does not stay held long. They are sent only if the queue is otherwise clear, so their ages are greater than regular TRs.

Note that UCCM is sending fewer images than FIFO sends. This is because the operator has the chance to preview the images and reject ones that do not appear to be useful. These images could have too many clouds, not show anything interesting, or be of poor quality. The FIFO radio cannot prevent those from being sent.

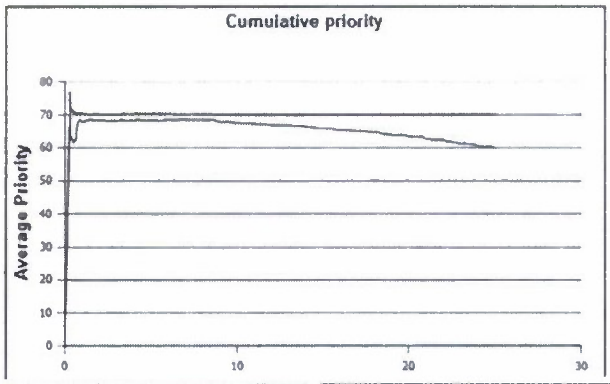
20:00

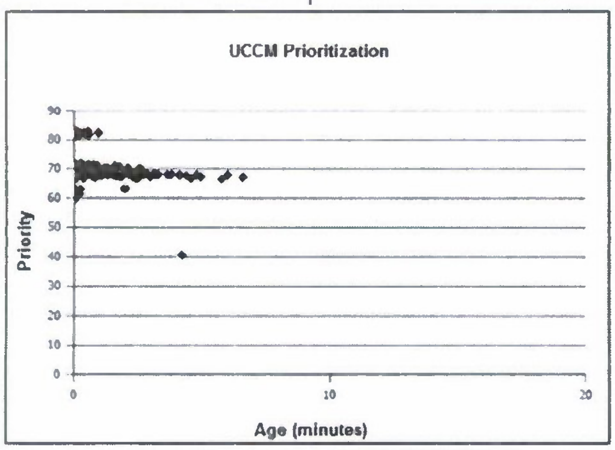


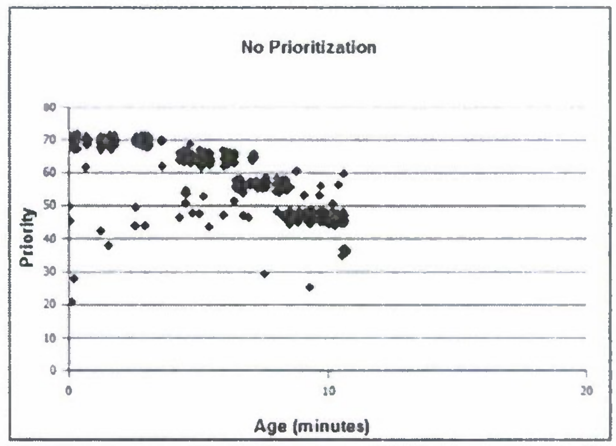
Enough of the mission has passed that aging is starting to make a difference. The average priority for FIFO starts to decline. Because it sends every image, the delays back up the rest of its TRs. By the time they get sent, these TRs are older than the corresponding UCCM TRs and have lower priority because of their age. After five minutes, an AIS or track will have had several more recent readings of the same entity, which have more value. The FIFO scatter chart mirrors the average priority. TRs that are delayed before being transmitted have lower priority.

Case I

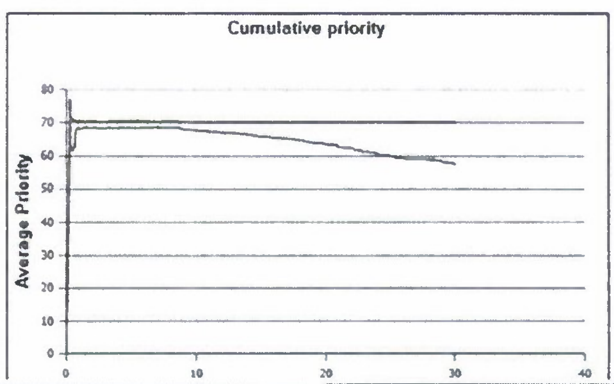
25:00

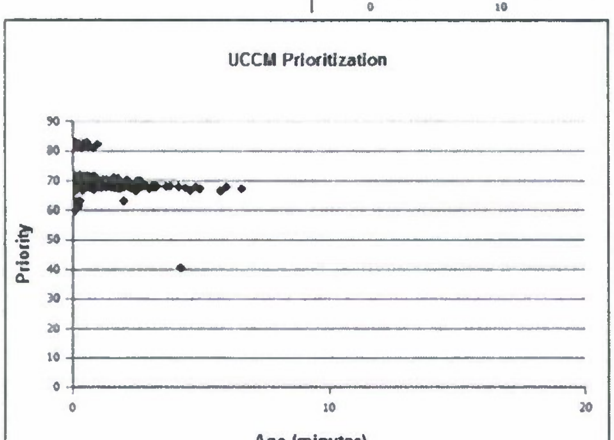


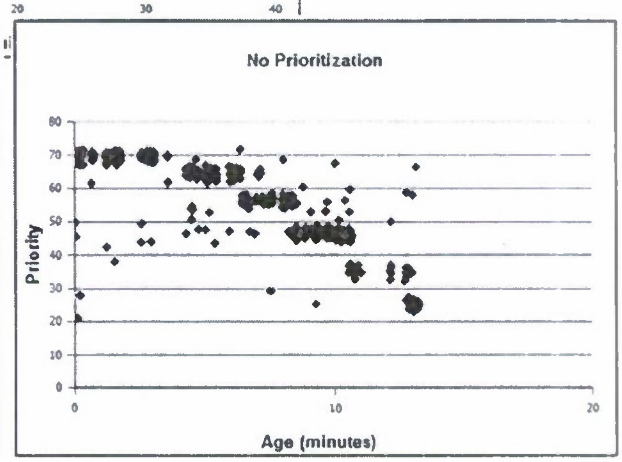




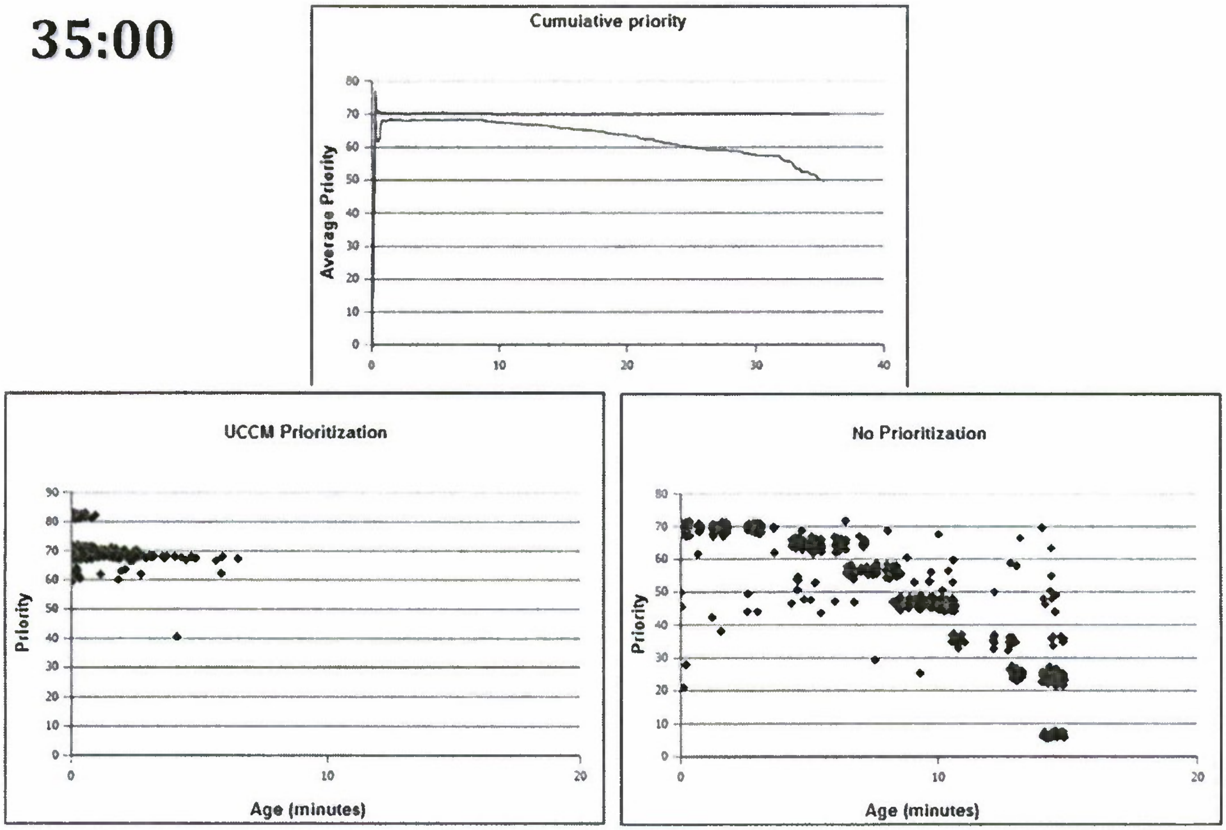
30:00



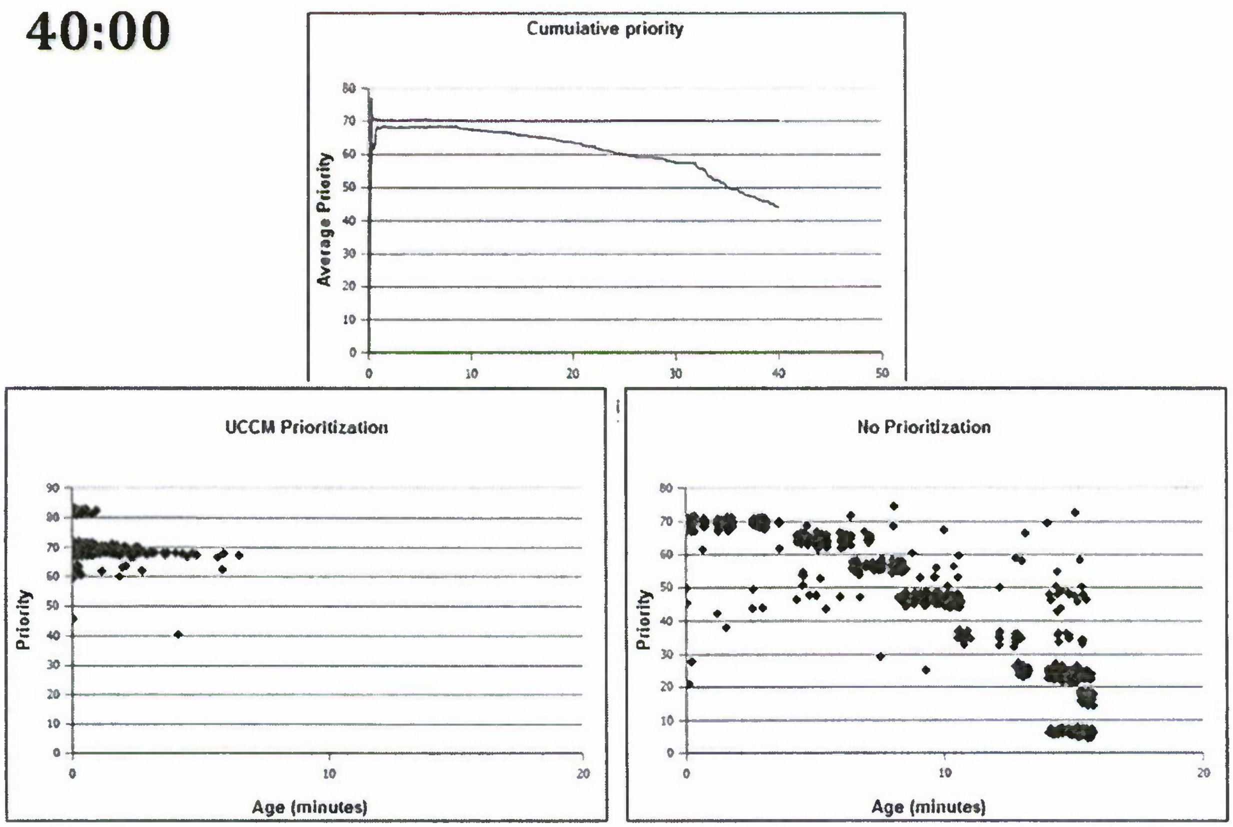




35:00

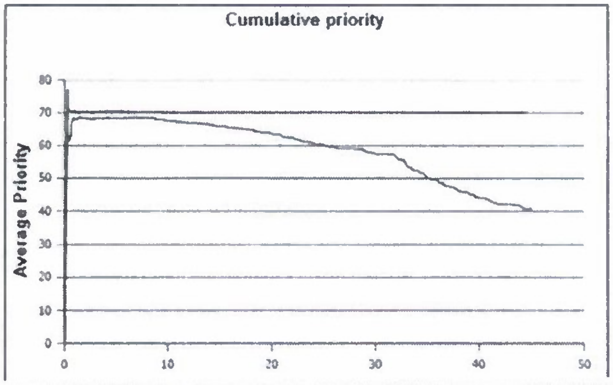

Cumulative priority
Average Priority
UCCM Prioritization
Priority
Age (minutes)
No Prioritization
Priority
Age (minutes)

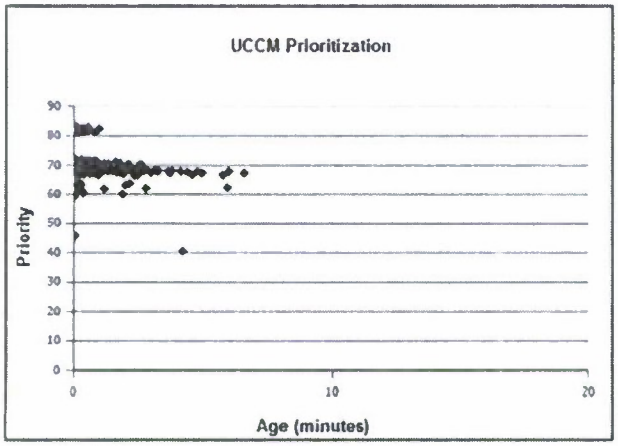
40:00

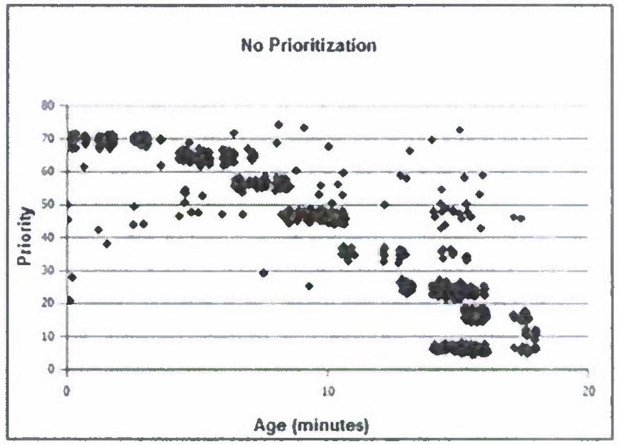

Cumulative priority
Average Priority
UCCM Prioritization
Priority
Age (minutes)
No Prioritization
Priority
Age (minutes)

Case I

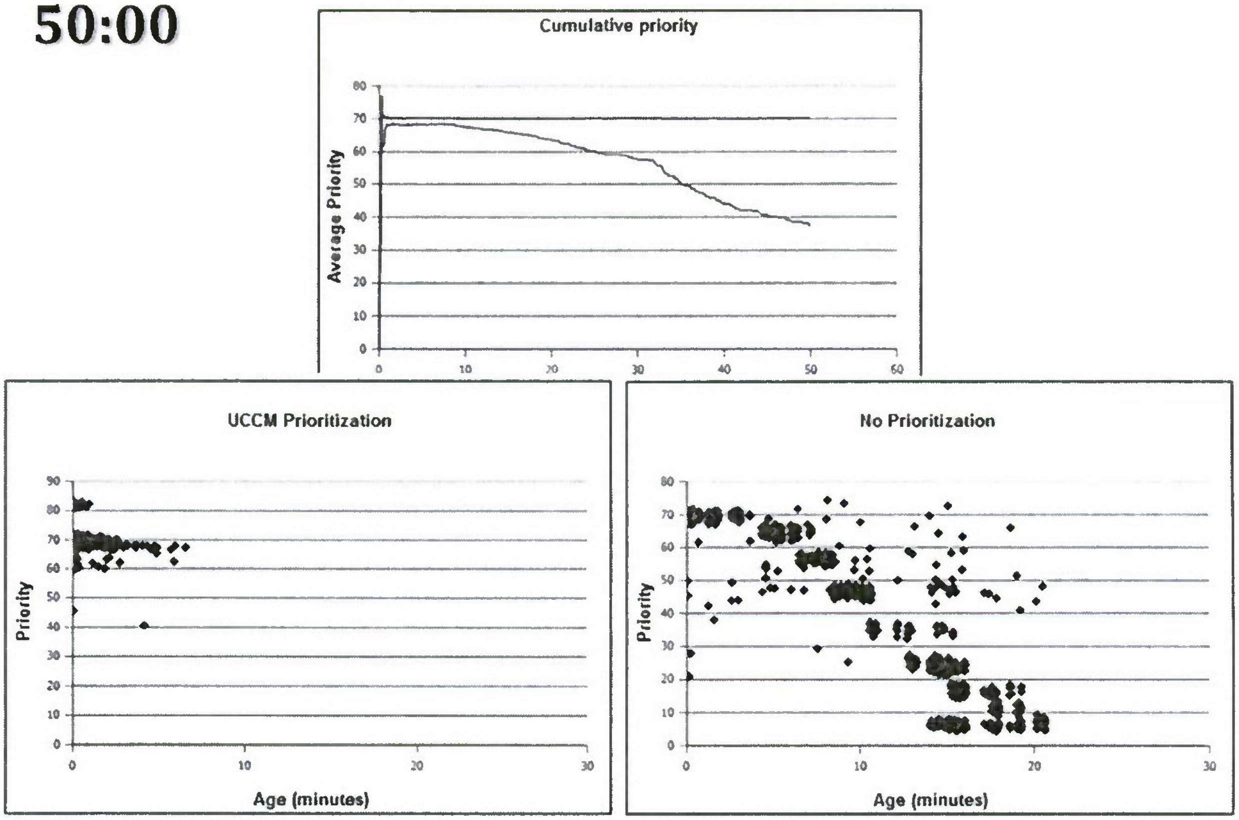
45:00


Cumulative priority
Average Priority
0
10
20
30
40
50
60
70
80
0
10
20
30
40
50

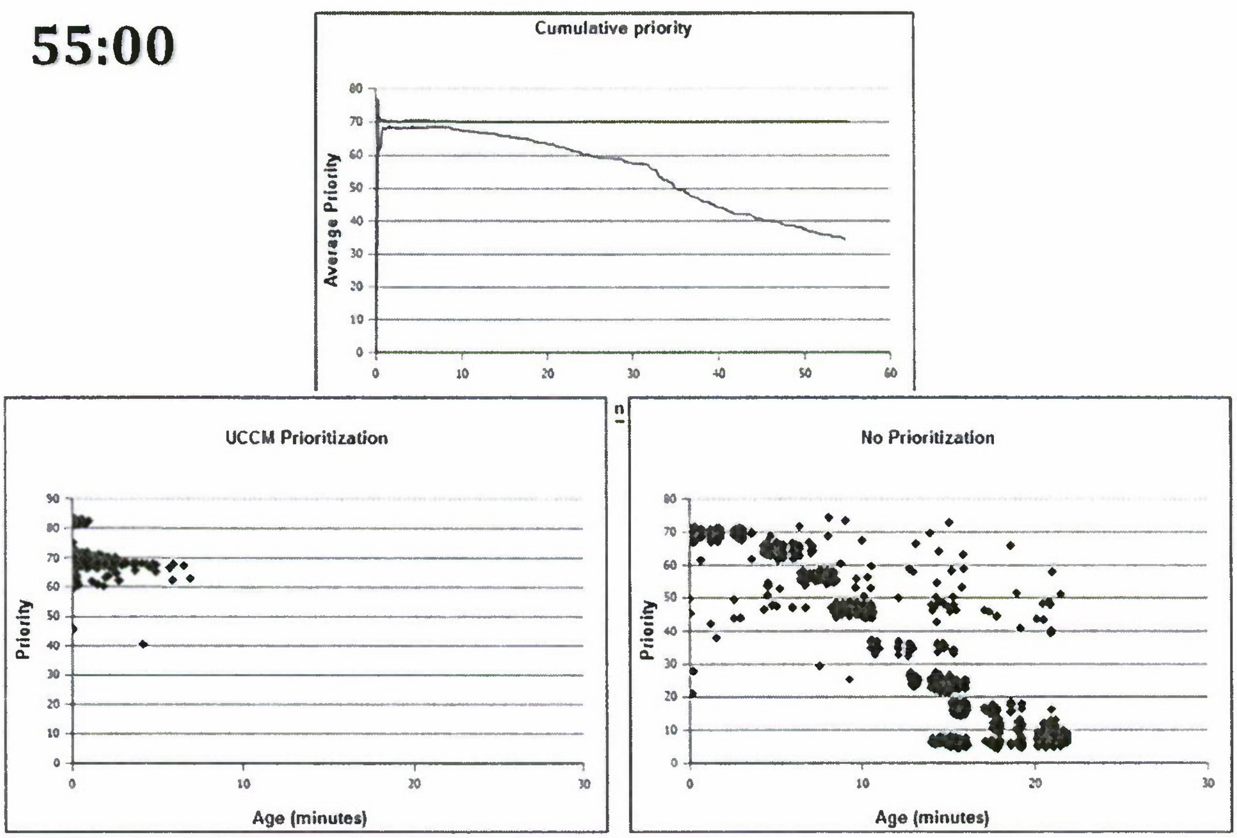

UCCM Prioritization
Priority
0
10
20
30
40
50
60
70
80
90
0
10
20
Age (minutes)


No Prioritization
Priority
0
10
20
30
40
50
60
70
80
0
10
20
Age (minutes)

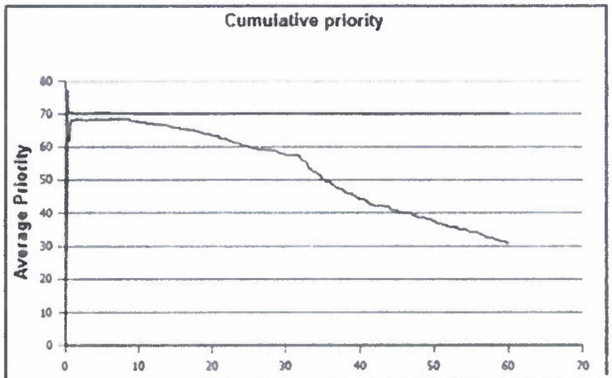
50:00

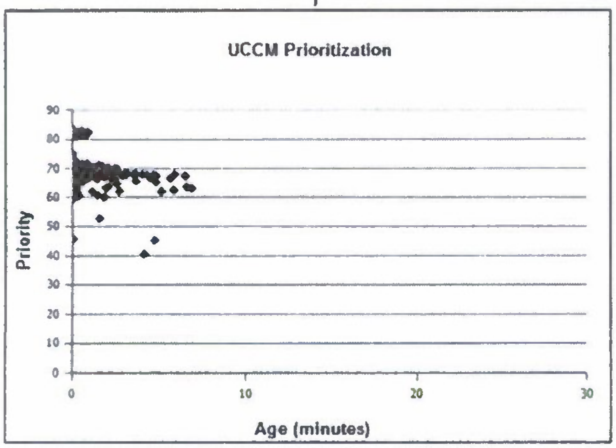

Cumulative priority
Average Priority
UCCM Prioritization
Priority
Age (minutes)
No Prioritization
Priority
Age (minutes)

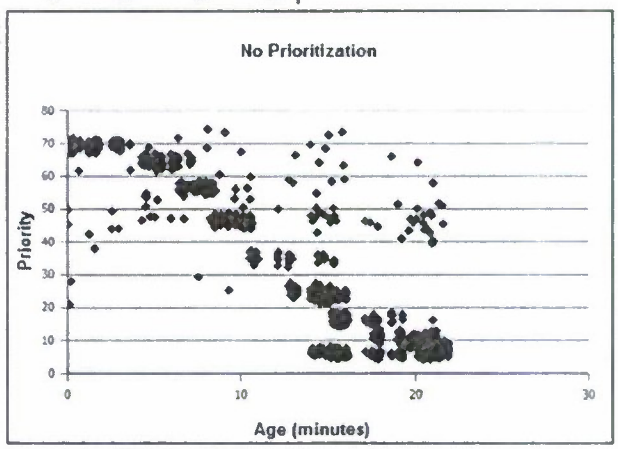
55:00


Cumulative priority
Average Priority
UCCM Prioritization
Priority
Age (minutes)
No Prioritization
Priority
Age (minutes)

60:00







An hour into the mission, the pattern is very clear. Even with good bandwidth, even a few images in the data stream will cause delays in the FIFO radio that lead to outdated small TRs being sent in preference to more current TRs. UCCM is able to decide to avoid sending certain low-value TRs, so it sends fewer overall. The ones it does send, get sent with little delay.

Metrics Page		Mission Elapsed Time: 01:00:06
The metrics show how well UCCM optimizes the information flow with priorities and how UCCM is reacting to circumstances.		
	UCCM	**FIFO**
TR arrived from the aircraft	7266	7266
TRs enqueued	2899	2856
TRs sent	2899	1818
TRs in the queue, ready to send	0	1037
TRs held	0	n/a
TRs with priority < minPriorityToSend	0	n/a
TRs deleted due to priority < minPriorityToKeep	0	n/a
TRs in transmission	0	1
Average priority of sent TRs	70.05	30.75
Average rules Per second	32.8474	
Average CPU utilization	44.39	

Figure 13: Case I Summary Metrics

Figure 13 shows that the UCCM radio was able to send about 1,000 more TRs than the FIFO radio did. These "missing" TRs remain in the FIFO radio's active queue, getting older and older as they wait to be sent. Bandwidth was relatively high, so UCCM didn't have to delete any of its TRs for excessive age. That a few images have such a large effect is underlined by the negabits metrics shown in Figure 13.

Number of TRs UCCM chose not to send	74
Number of TRs sent	2899
Size of data not sent (MB)	132.555
Size of data sent (MB)	244.901
Negabits ratio	0.54
Average radio output rate (Kbps)	556.37

Figure 14: Case I Negabits Metrics

74 TRs out of 2,899 made the difference. However, they were all images so for every 2 MB that UCCM did send, it was able to avoid sending 1 MB. This is underlined again by the histograms of TRs sent by type. Only EO and IR images were deleted. It appears that only a small number of SAR images were generated, and the operator must have accepted them all.

Case I

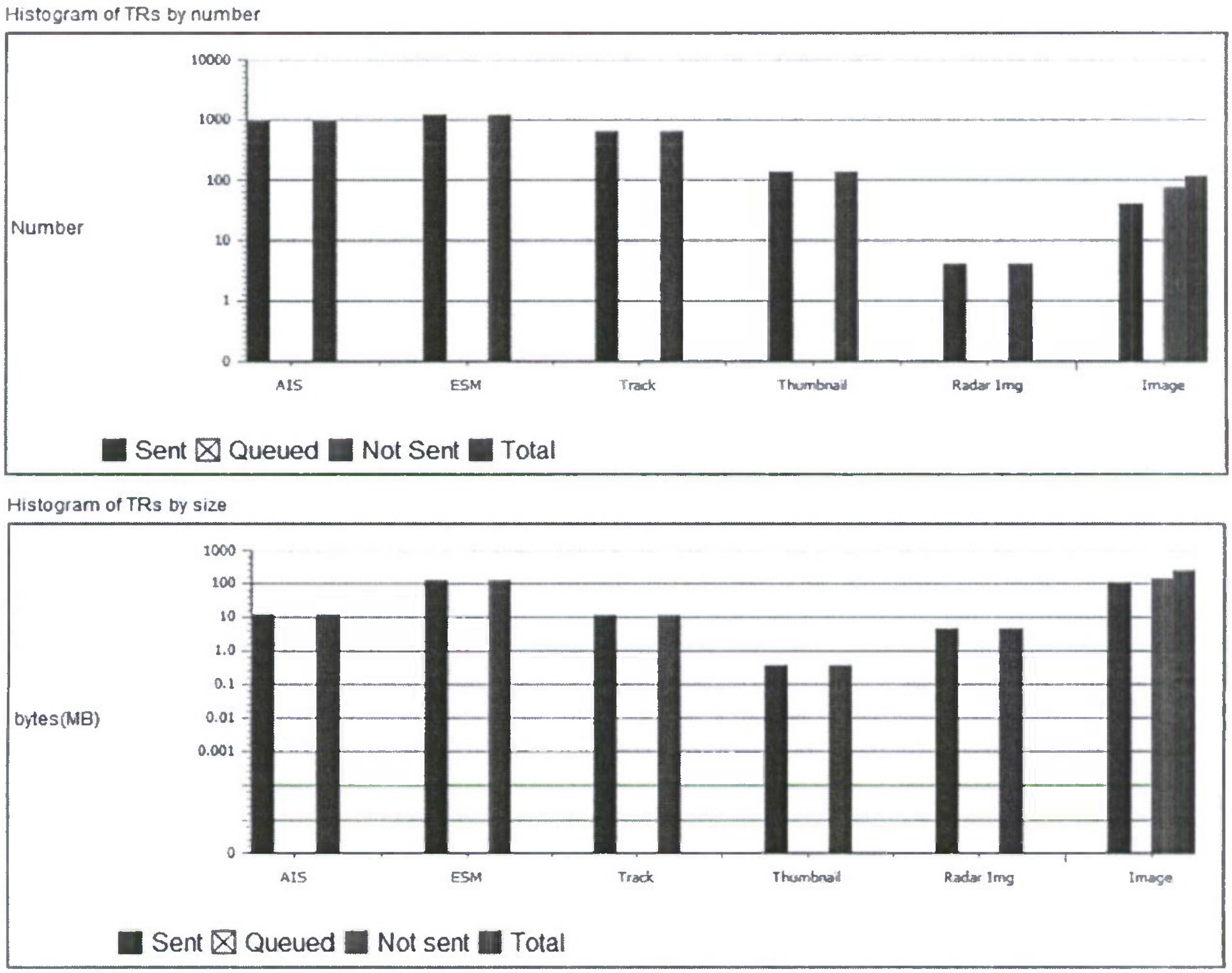


Figure 15: Case I Statistics on the Types of TRs that were Sent

Case II: Insufficient Bandwidth

This case shows UCCM under bandwidth stress. As with the last case, all sensors output to the same Ku channel. There is also a higher proportion of images, with the same 40-40-20 selection ratio. Later in the mission, we let the bandwidth increase to prevent the FIFO queue from backing up excessively. The bandwidth profile is shown in Figure 16. As in Case I, the case is presented as a set of snapshots of metrics pages, collected every five minutes.

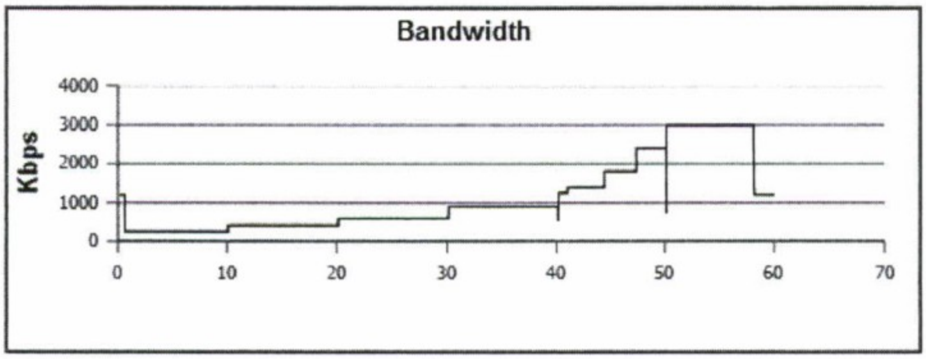
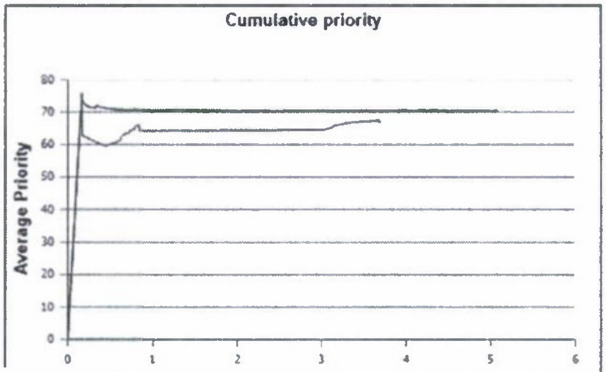
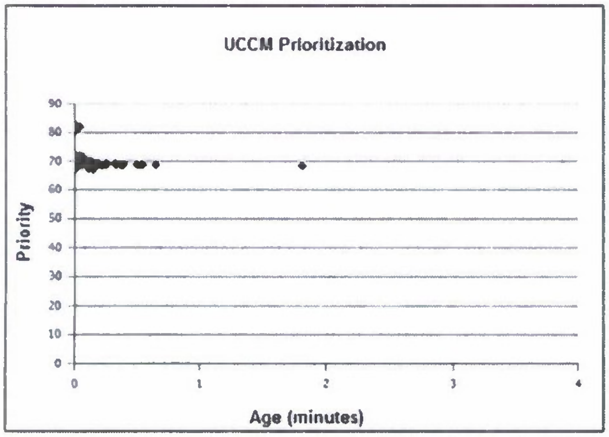
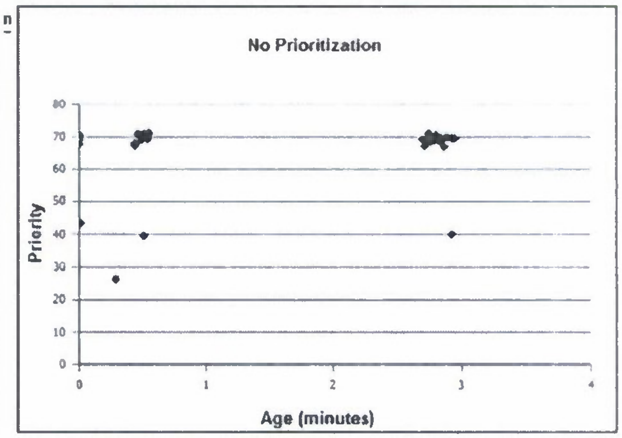


Figure 16: Case II Bandwidth Profile

5:00

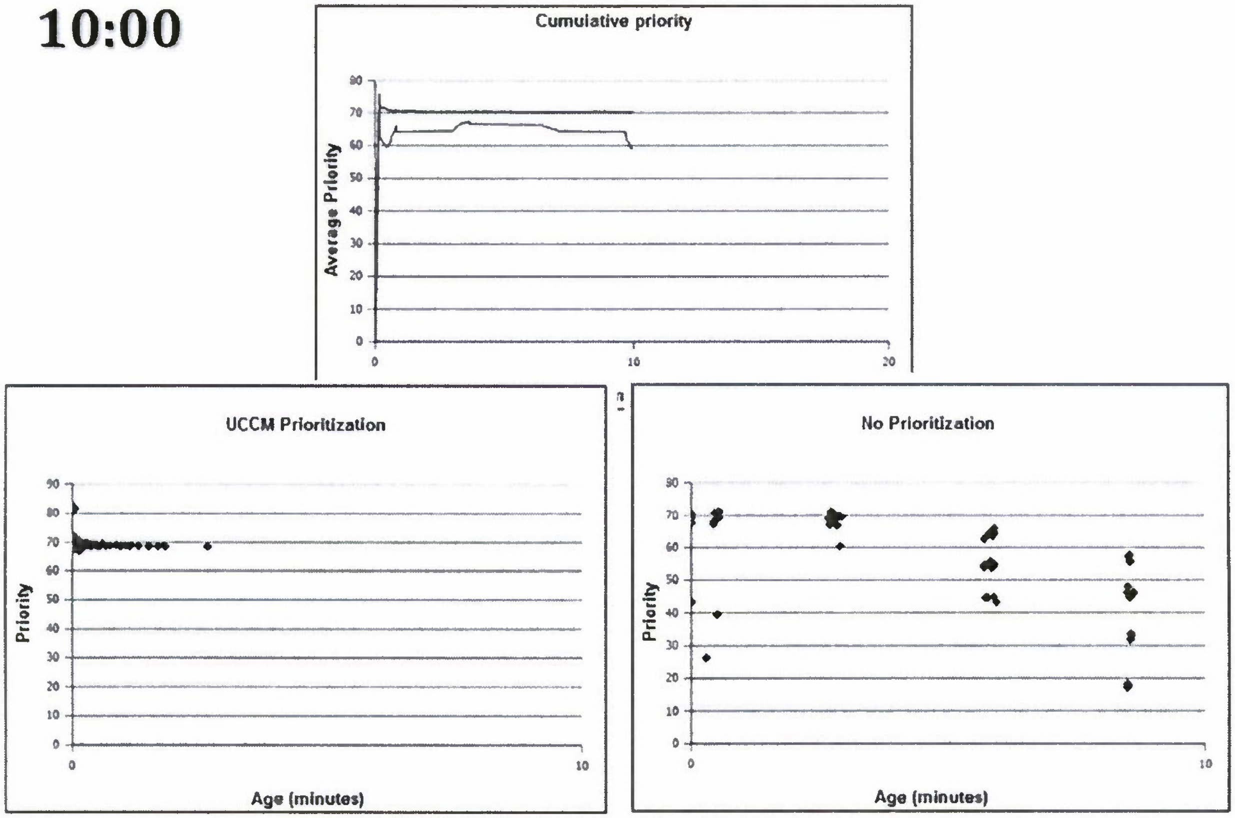




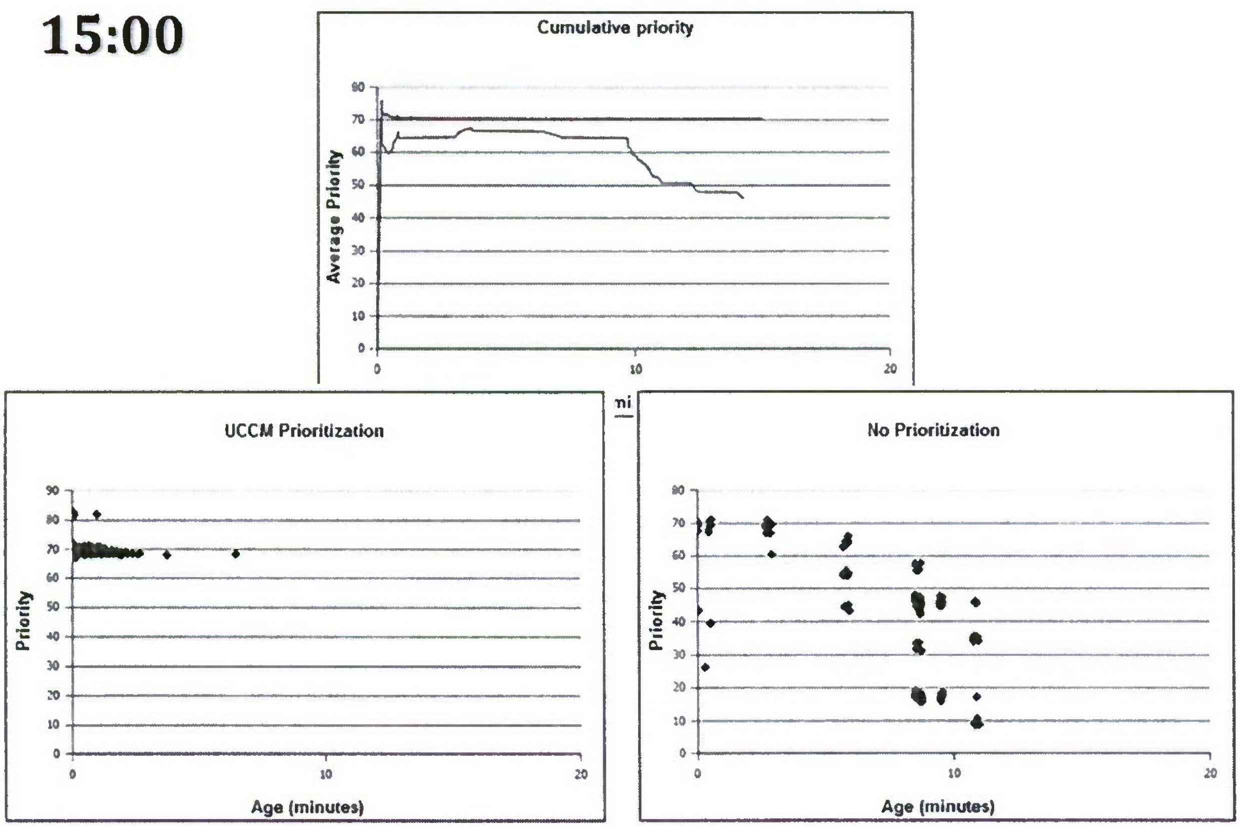


The FIFO average priority is moderately lower than in the previous case. This is not significant this early in the mission. Looking at the FIFO scatter chart, it seems that two or three large images have distorted the average.

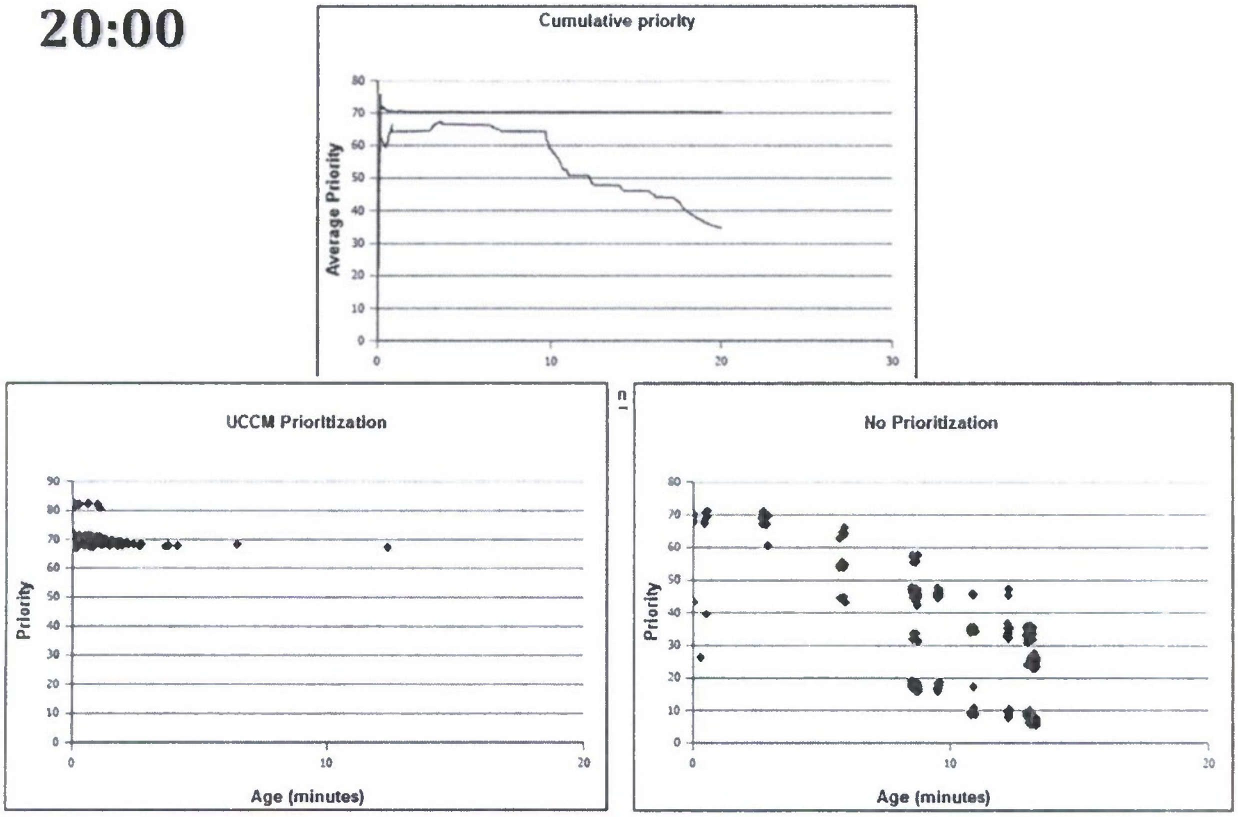
10:00


Cumulative priority
Average Priority
UCCM Prioritization
Priority
Age (minutes)
No Prioritization
Priority
Age (minutes)

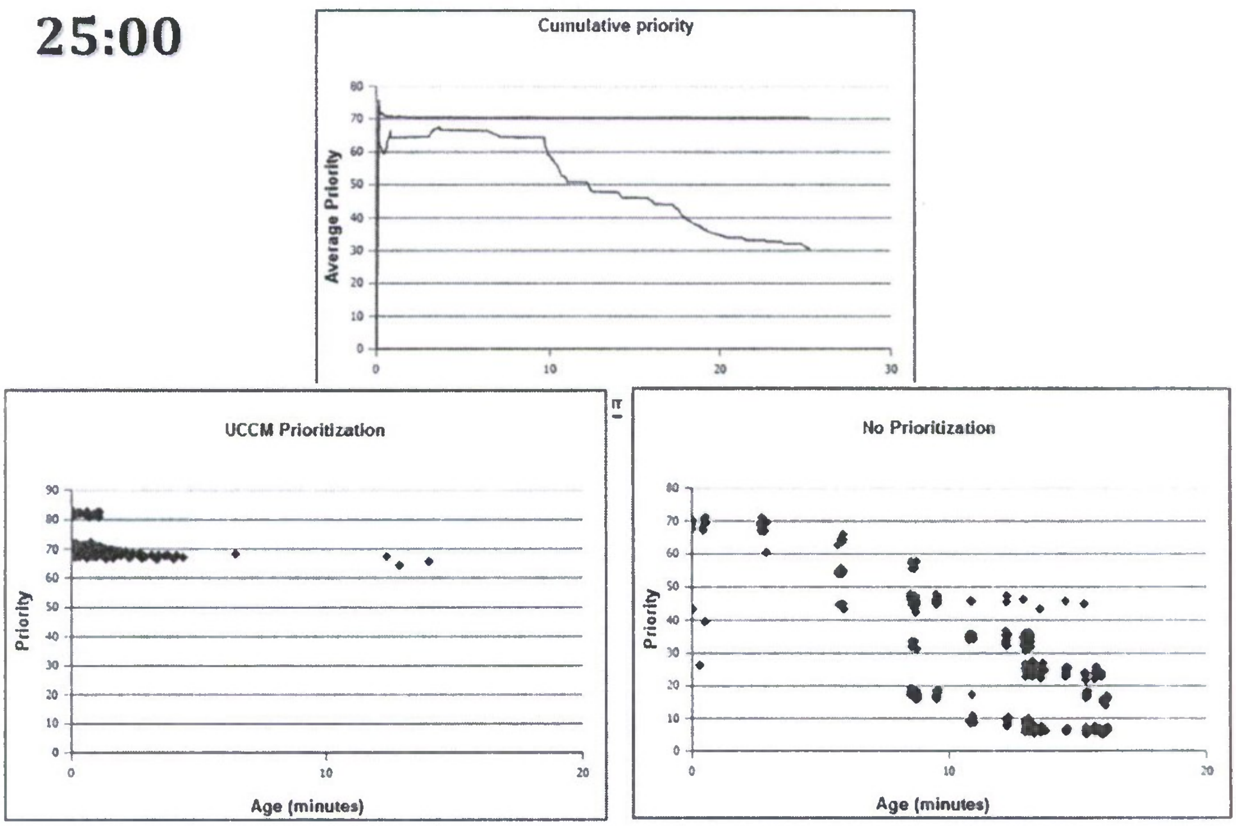
15:00


Cumulative priority
Average Priority
UCCM Prioritization
Priority
Age (minutes)
No Prioritization
Priority
Age (minutes)

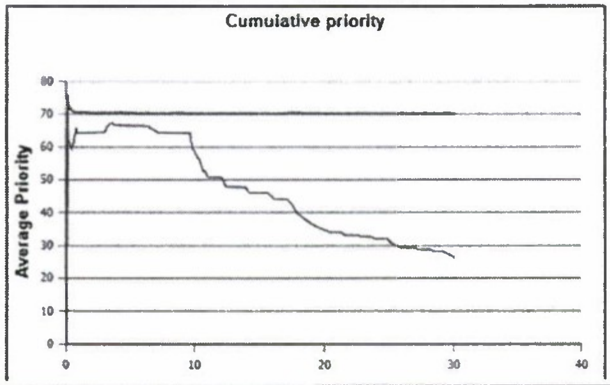
20:00

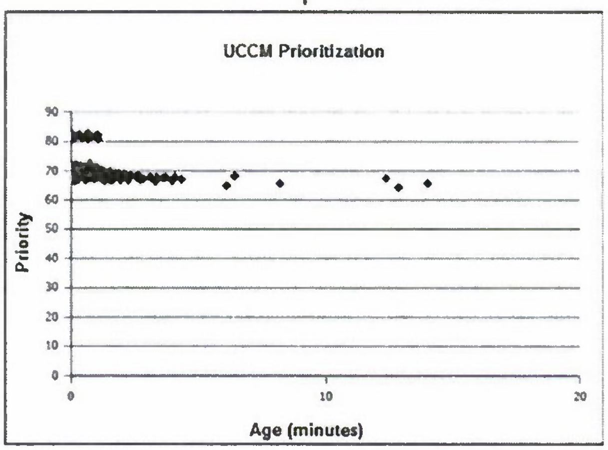

Cumulative priority
Average Priority
UCCM Prioritization
Priority
Age (minutes)
No Prioritization
Priority
Age (minutes)

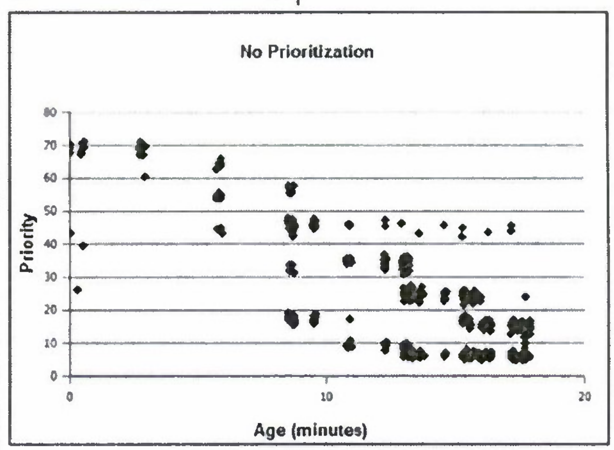
25:00


Cumulative priority
Average Priority
UCCM Prioritization
Priority
Age (minutes)
No Prioritization
Priority
Age (minutes)

30:00



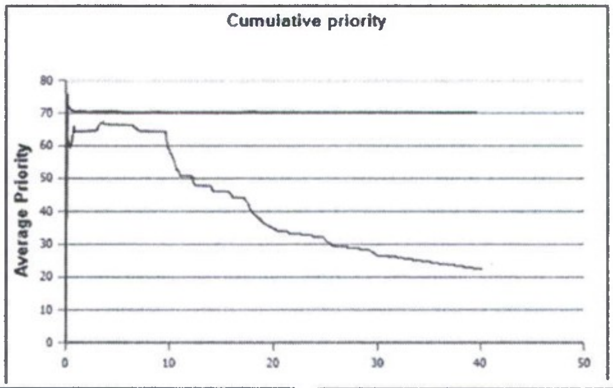


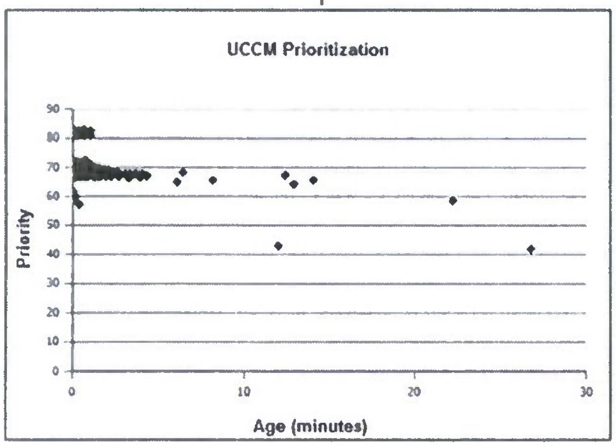


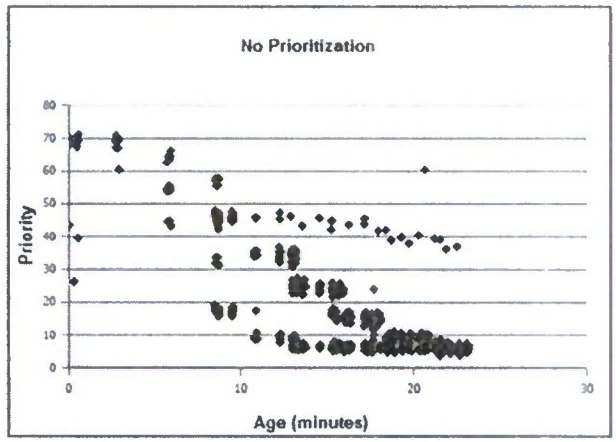
The bandwidth has remained generally less than the sensors' output rate. The FIFO queue is backing up at an alarming rate because TRs arrive faster than they can be sent. The impacts of aging are more visible because there are more TRs to age, and they are staying in the queue longer than in Case I. We are starting to see differential aging between ESM and AIS (which are on the quick-death aging curve) and tracks (which are on the fastest aging curve). The difference can be seen in the two arcs starting from between about (5min, 65) and the lower-right corner.

There is an opportunity to refine the UCCM prioritization. It appears UCCM only sent a few small (therefore relatively high-priority) images. Should it have sent more? A few might also have been held, which will not show up in these charts.

40:00



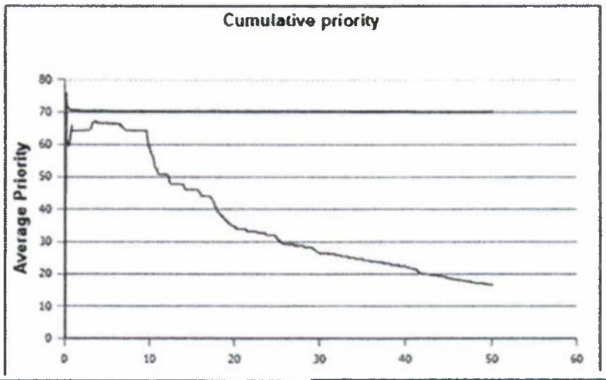


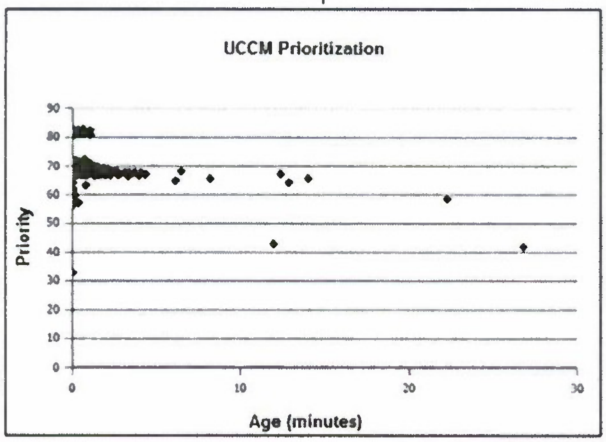


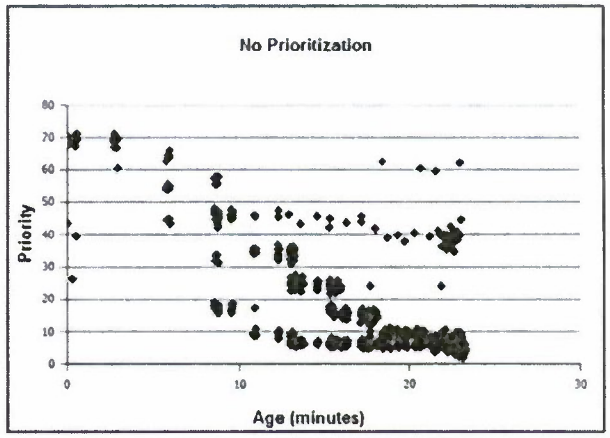
Some images are now appearing on the UCCM side. An old TR with a priority that does not reflect aging is generally something that has been held, and has now been released.

Case II

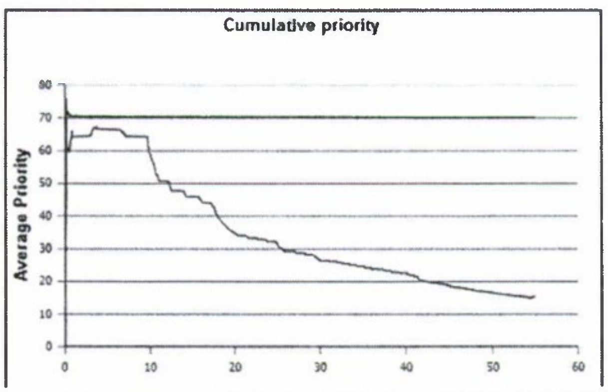
50:00

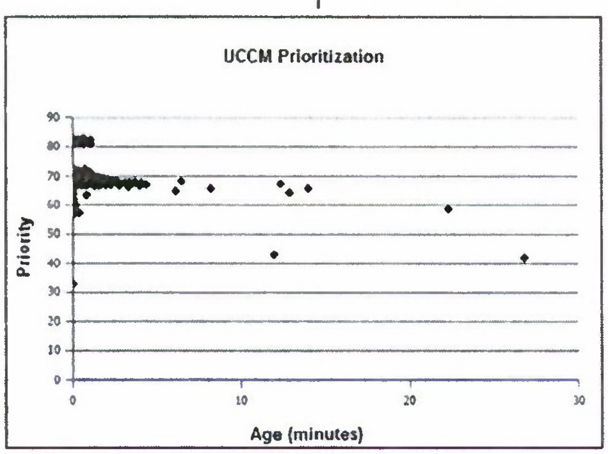


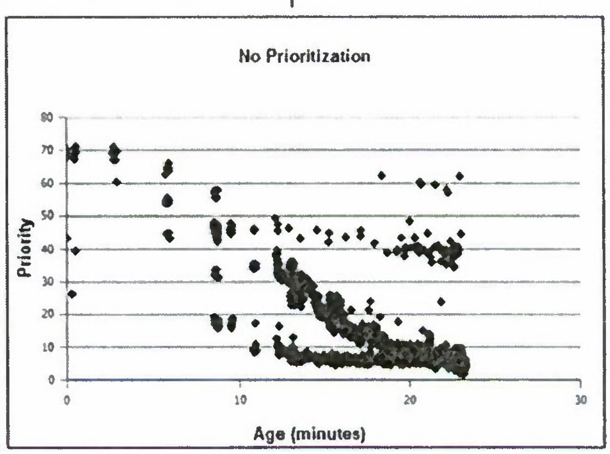




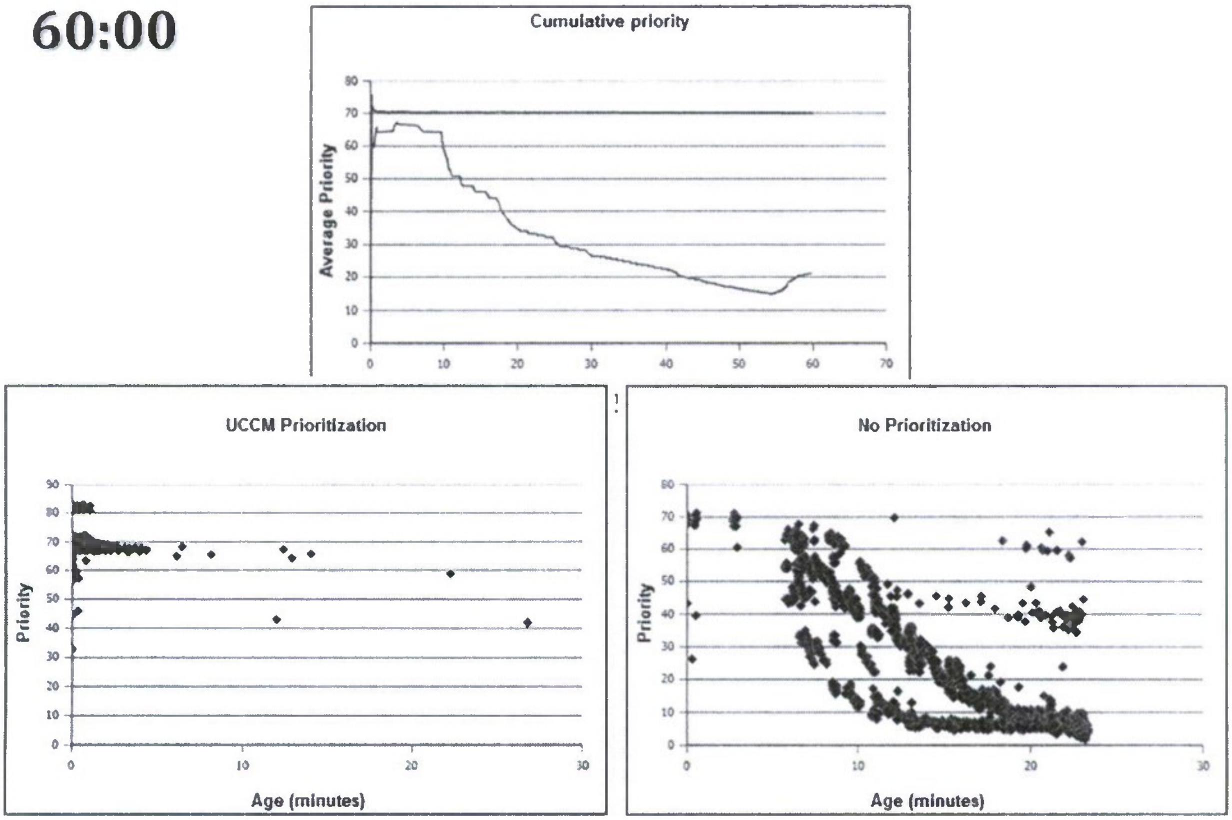
55:00







60:00



It is very interesting to compare the previous snapshot to this one (55:00 and 60:00 minutes). Bandwidth has been increasing in the last 20 minutes but the very large backlog prevented that improvement from helping FIFO very much. In the last 5–10 minutes bandwidth jumped up to 3,000 Kbps, set very high to see what "infinite" bandwidth would do. Note that the FIFO chart filled in a lot of TRs with age between 5 and 12 minutes. It has cleared a lot of its queue. And because younger TRs are being sent, they have not aged as much, their priority is higher, and the FIFO cumulative priority curve is actually turning upward.

The other item of note is that all three aging curves are now represented. Images age much more gradually than other types of TRs. So we now see the lower curve of elderly tracks, then a curve of AIS and ESMs, and finally a thinner curve of images. The image curve actually increases for 10 minutes, and after 20 minutes is back to the same value as age = 0. This is why the image curve is close to horizontal.

This case is a good example of why the negabits metric is very contextual. Since bandwidth became very high at the end, almost everything on the queue was sent. Negabits in Figure 17 is then around zero. If we had captured negabits at 30 or 40 minutes, there would have been a more marked difference. The final metrics screen (Figure 18) also reflects the period of high bandwidth.

Number of TRs UCCM chose not to send	67
Number of TRs sent	3389
Size of data not sent (MB)	18.326
Size of data sent (MB)	296.822
Negabits ratio	0.06
Average radio output rate (Kbps)	674.74

Figure 17: Case II Negabits Results

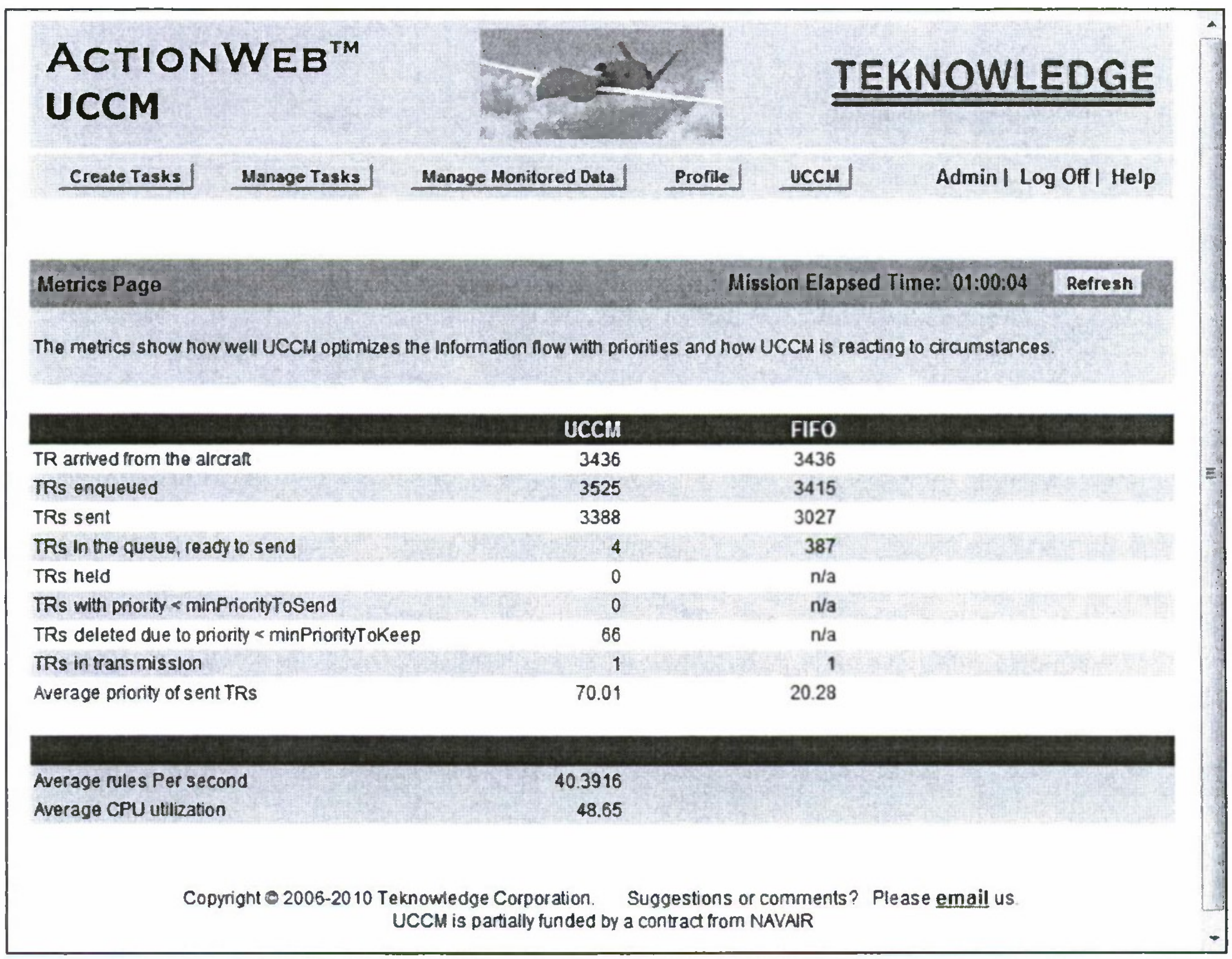


Figure 18: Case II Summary Metrics

Case III: Handling of Special Cases

This case highlights UCCM's ability to treat special cases differently from routine TRs, and to honor the operator's intent. The bandwidth profile and mission scenario are shown in Figure 19.

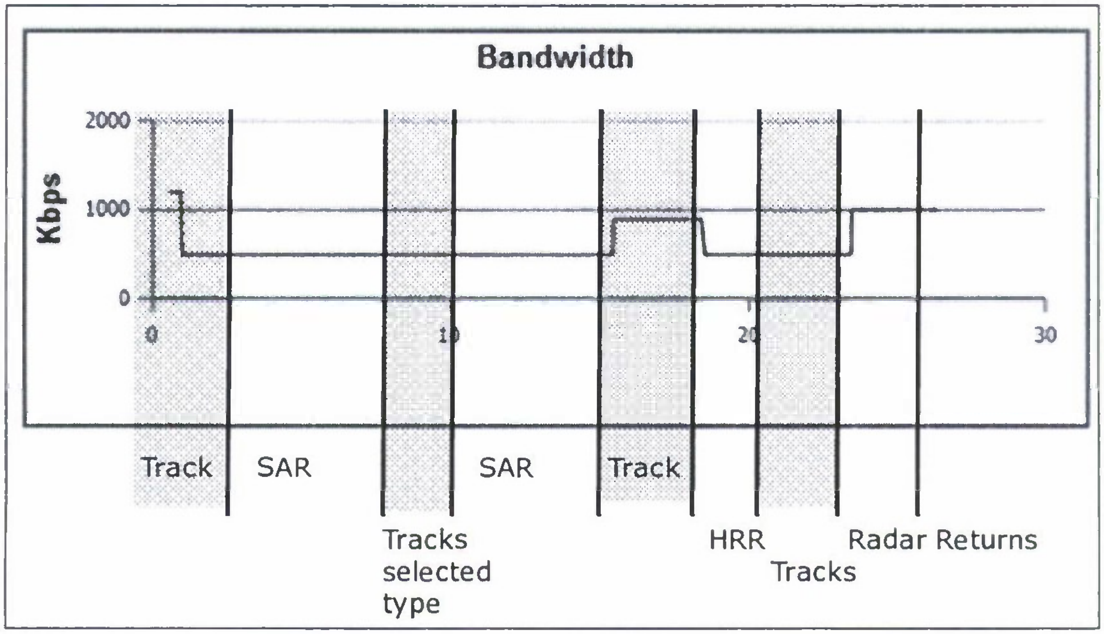


Figure 19: Case III Bandwidth Profile

The shaded sections and the changes in the radar mode reflect the following mission:

- The operator starts in routine mode. The radar produces tracks by default.
- The operator then decides to collect some SAR imagery. They do not want to let the tracks get too out of date, so switches back to tracks for a few minutes during SAR collection. In order to ensure that these updates do get sent, the selectedType for this channel is set to tracks.
- After looking at the SAR images briefly, the operator goes back for some quick HRR collection to follow up. This is an unusual mode, so UCCM should give it priority.
- There is still something that doesn't quite make sense, so the operator switches the radar to ranging mode and views a few minutes of raw returns. This radar mode is most useful when viewed in real time, or close to that. UCCM gives these TRs special priority to make sure this happens.

Sort By: Priority Descending Refresh

TR Id	Type	Priority	Size (KB)	Age	Radio Status	AT Radio Status
200079	Thumbnail	83.14	2	00:00:00	Sent	
200479	Thumbnail	83.09	2	00:00:00	Sent	
200579	Thumbnail	82.98	2	00:00:00	Sent	
200959	Thumbnail	82.96	1	00:00:01	Sent	
201659	Thumbnail	82.88	1	00:00:01	Sent	
201859	Thumbnail	82.87	1	00:00:01	Sent	
200379	Thumbnail	82.84	2	00:00:01	Sent	
201159	Thumbnail	82.69	1	00:00:01	Sent	
200679	Thumbnail	82.66	2	00:00:00	Sent	
201559	Thumbnail	82.58	1	00:00:16	Sent	
201759	Thumbnail	82.56	1	00:00:01	Sent	
200779	Thumbnail	82.48	2	00:00:05	Queued	
201359	Thumbnail	82.34	1	00:00:00	Sent	
201259	Thumbnail	82.28	1	00:00:02	Sent	
200089	Thumbnail	82.21	4	00:00:01	Sent	
201459	Thumbnail	81.87	1	00:00:11	Sent	
200279	Thumbnail	81.65	2	00:00:00	Sent	
201059	Thumbnail	81.62	1	00:00:00	Sent	
200179	Thumbnail	81.55	2	00:00:13	Sent	
200859	Thumbnail	81.34	1	00:00:01	Sent	
135	SAR image	73.69	606	00:00:06	Sent	
54	Track	72.02	10	00:00:29		Sent
54	Track	72.02	10	00:00:05	Sent	
144	Track	71.83	13	00:00:20		Sent

All tracks have priority < 73

Figure 20: Case III at End of First SAR Collection

Figure 20 is taken from the screen that shows TRs sent by UCCM. It is sorted by priority for this image. All thumbnails have greater priority than all tracks. UCCM is just starting to get SAR images sent out. Others are waiting on the queue.

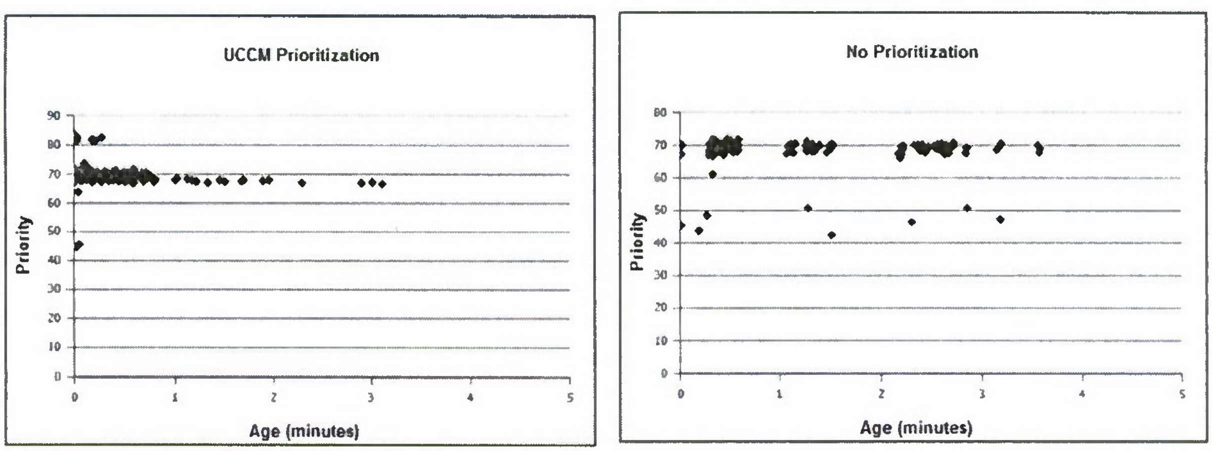
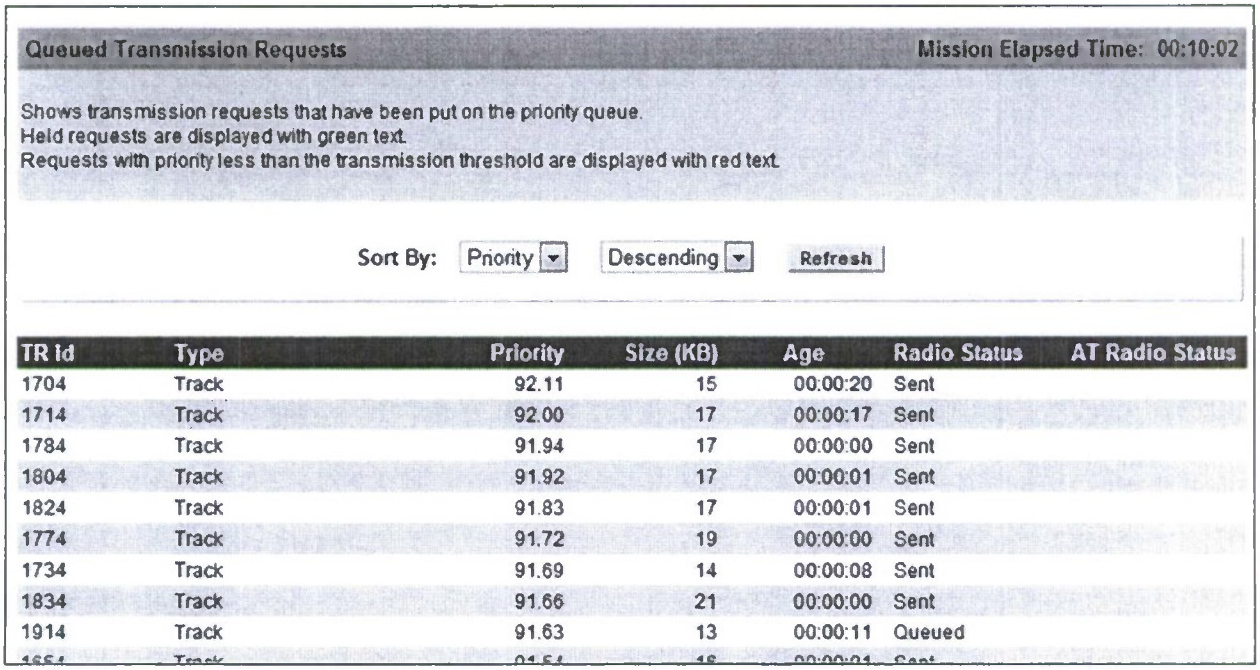


Figure 21: Case III TRs At 8:00 Minutes



Queued Transmission Requests — Mission Elapsed Time: 00:10:02

Shows transmission requests that have been put on the priority queue.
Held requests are displayed with green text.
Requests with priority less than the transmission threshold are displayed with red text.

Sort By: Priority | Descending | Refresh

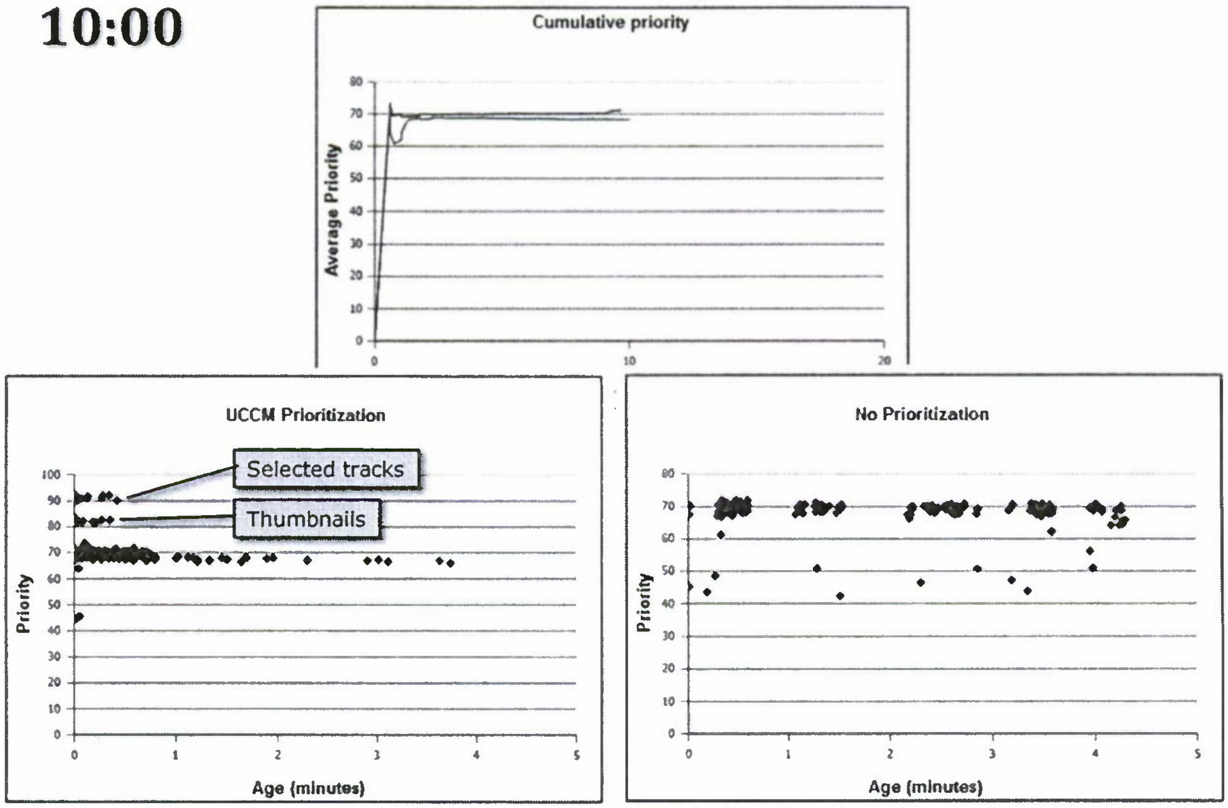
TR Id	Type	Priority	Size (KB)	Age	Radio Status	AT Radio Status
1704	Track	92.11	15	00:00:20	Sent	
1714	Track	92.00	17	00:00:17	Sent	
1784	Track	91.94	17	00:00:00	Sent	
1804	Track	91.92	17	00:00:01	Sent	
1824	Track	91.83	17	00:00:01	Sent	
1774	Track	91.72	19	00:00:00	Sent	
1734	Track	91.69	14	00:00:08	Sent	
1834	Track	91.66	21	00:00:00	Sent	
1914	Track	91.63	13	00:00:11	Queued	

Figure 22: Case III After Sending the Selected-Type Tracks

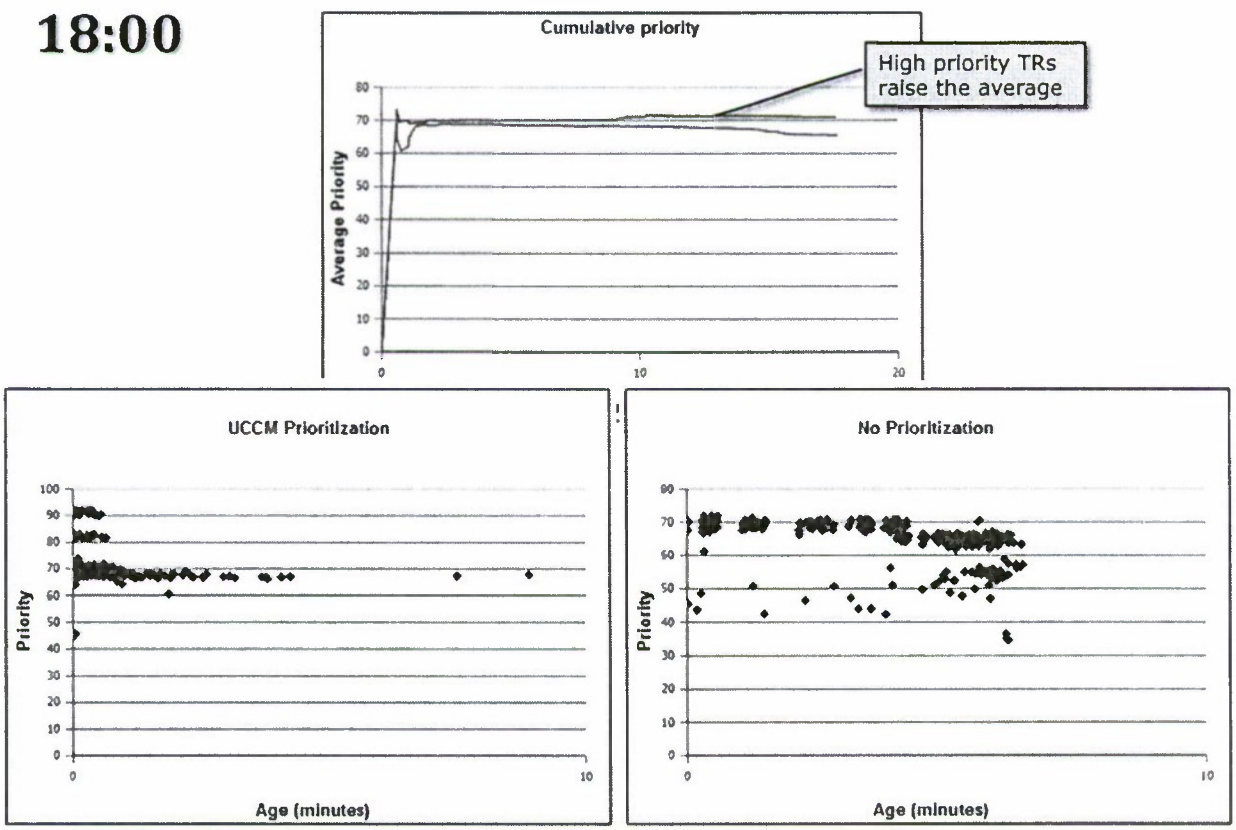
Since tracks are the selectedMode, they get a priority boost. The operator specifically said they want tracks, so the operator gets them. All of these tracks have priority in the 90s, while regular tracks have priority in the low 70s. Figure 22 shows that they have even have higher priority than thumbnails, and are currently at the top of the priority queue.

Case III

10:00



18:00



While the UCCM average priority is mostly dominated by a large number of small TRs with priority around 70, this case has a number of special-treatment TRs in the 90s. This raises the average priority.

The FIFO scatter chart shows the usual image-delay patterns. SAR images are smaller on average than IR or EO images. It looks like the cluster at 5 minutes and mid 50s consists of SAR images rather than aged TRs.

1894	Track	89.89	21	00:00:31	Sent	
1874	Track	89.82	25	00:00:01	Sent	
100120	HRR image	89.31	12	00:02:17		Queued
100180	HRR image	87.95	11	00:00:15		Queued
100160	HRR image	87.84	14	00:00:52		Queued
120	HRR image	87.80	12	00:00:00	Sent	
180	HRR image	86.83	11	00:00:02	Sent	
160	HRR image	86.81	14	00:00:07	Sent	
130	HRR image	86.69	19	00:00:00	Sent	
170	HRR image	86.42	17	00:00:08	Sent	
100130	HRR image	85.47	19	00:01:56		Queued
150	HRR image	85.18	29	00:00:01	Sent	
140	HRR image	84.93	36	00:00:01	Sent	
100140	HRR image	84.59	36	00:01:40		Queued
201479	Thumbnail	83.16	2	00:00:00	Sent	
200079	Thumbnail	83.14	2	00:00:00	Sent	

Figure 23: Case III at 18:00. HRR Collection Just Ended

HRR images are small, as images go, and get a boost from the intent factor. If an operator asks for HRR, they really mean it and want to see them ASAP. Their priority ends up as higher than thumbnails, but not quite as high as specially selected tracks.

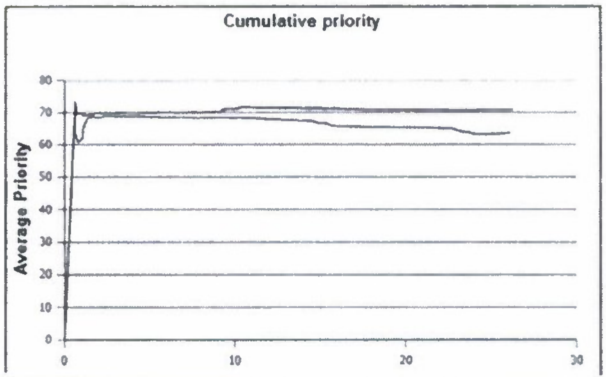
Case III

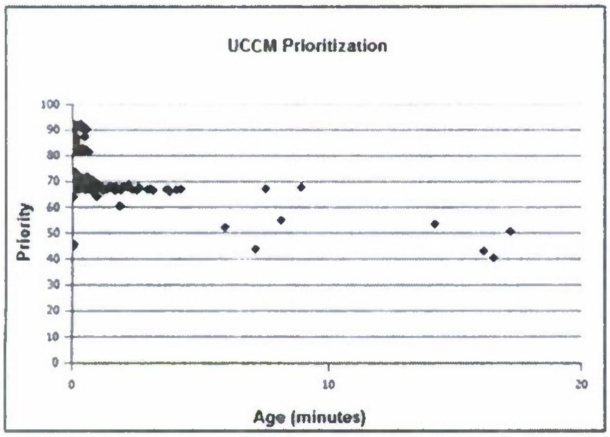
130	HRR image	86.69	19	00:00:00	Sent	
170	HRR image	86.42	17	00:00:08	Sent	
113853	Radar return	86.24	12	00:01:14		Queued
13903	Radar return	85.88	19	00:00:00	Queued	
13893	Radar return	85.84	16	00:00:00	Sent	
13843	Radar return	85.66	12	00:00:00	Sent	
13913	Radar return	85.60	23	00:00:00	Queued	
113793	Radar return	85.57	17	00:02:21		Queued
13873	Radar return	85.54	9	00:00:00	Sent	
13773	Radar return	85.49	14	00:00:08	Sent	
113833	Radar return	85.35	20	00:01:46		Queued
13853	Radar return	85.29	12	00:00:00	Sent	
150	HRR image	85.18	29	00:00:01	Sent	
113883	Radar return	85.13	9	00:00:17		Queued

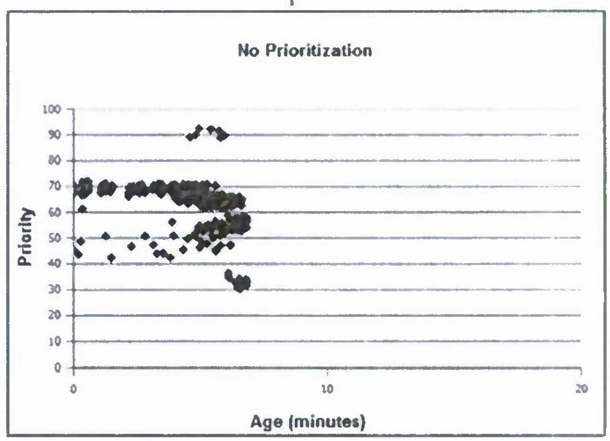
Figure 24: Case III at 25:00 Minutes. Collected Radar Returns

Figure 24 illustrates another special case. Like HRR, the operator won't switch the radar to this mode unless they really mean it. HRRs are small images, and radar retuns are bundles of readings. They overlap in size ranges, both get similar Intent boosts, so their priorities overlap.

25:00







The results from UCCM special handling are in:

- Selected mode tracks form the cluster in the 90s.
- A mixture of HRR images and radar returns occupy the cluster in the high 80s.
- The standard set of thumbnails occupies the low 80s.
- There is a sprinkle of IR and EO images thereafter.

This case only ran 30 minutes, so aging is just starting to make an impact on the FIFO side. We do see a pattern from an increased number of images of various types.

There is only a modest difference in cumulative average priority, but that is not the whole story. Bandwidth was high enough that most TRs got sent. The difference is that UCCM was able to rearrange the order in which things were sent according to the needs of the operator. Most of the FIFO TRs took 5 minutes before they were sent, and all of the special-handling TRs needed to be sent immediately. UCCM was able to do that.

The UCCM scatter chart shows some images taking 10–20 minutes before being sent. These are the larger IR and EO images. The radio has a lot of high-priority items that get sent first, and then the usual mix of small TRs. All that special handling means the lower priority TRs will be delayed more than usual, as the price of the rush priority for other TRs.

Oct 15, 2010

Defense Technical Information Center (DTIC)
Attn: Document Acquisition (DTIC-OCA)
8725 John J. Kingman Rd - Ste 0944
Ft. Belvoir, VA 22060-6218

Dear Sir or Madam,

We just completed Phase II of the User-Centered Communications Manager SBIR. This is NAVAIR contract N68335-08-C-0493 to Teknowledge Corporation.

In accordance with DFAR 252.235-7011, the contractor shall deliver 2 additional copies of the final report under this contract to the address below. Please find these two copies enclosed.

Thank you.
Dr. Allan Terry
Principal Investigator
aterry@teknowledge.com